

# Import external zoning on yield data

*B. Charnomordic*

*2017-09-12*

## Contents

Import shape1 object 4

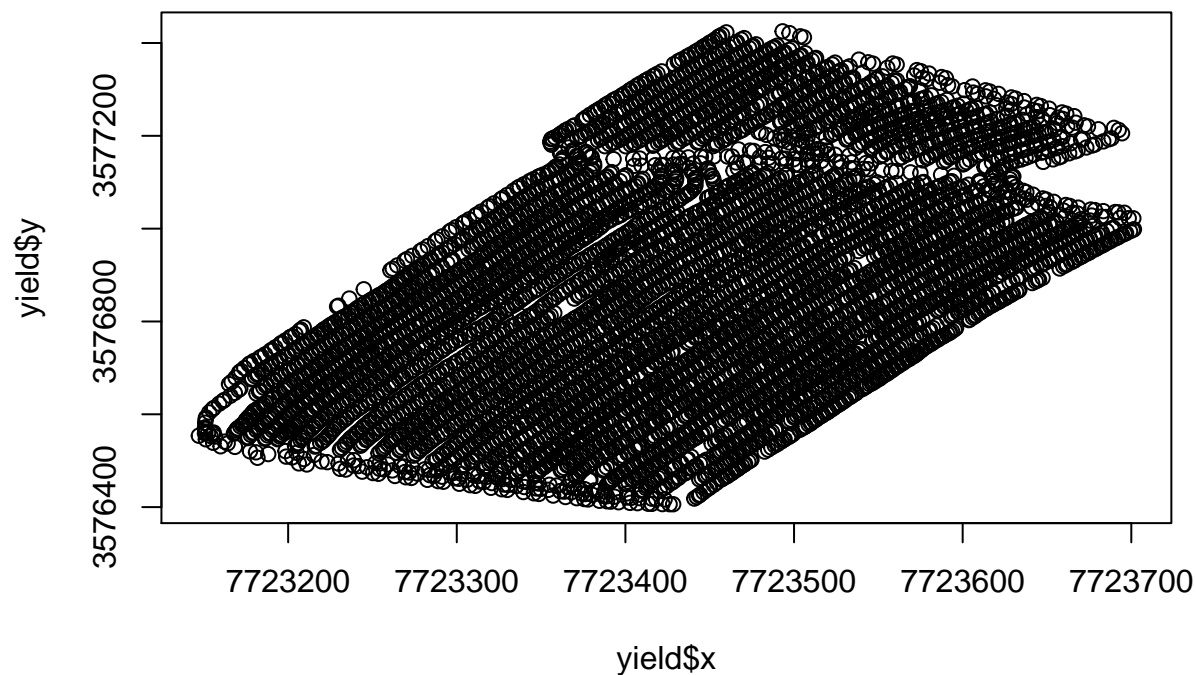
Session informations 6

```
library(geozoning)
library(ggplot2)
library(sp)
```

This vignette reads an existing zoning, that was not done with geozoning, in order to calculate its quality criterion. It uses 2 objects from the geozoning package. The shape1 object contains the zoning, it is a SpatialPolygonsDataFrame, where each SpatialPolygons is a zone. It was read from a shapefile, available in the extdata directory. Yield contains yield data, imported from a csv file, also available in the extdata directory.

View yield data

```
data(yield)
plot(x=yield$x, y=yield$y)
```



Draw

manual boundary (could also be read from a shape file)

```
boundary0=list()
boundary0$x=c(7723128,7723223,7723283,7723376,7723430,7723453,7723605,7723713,7723699,7723638,7723696,
              7723562,7723531,7723494,7723459,7723344,7723356,7723221,7723158,7723129)
```

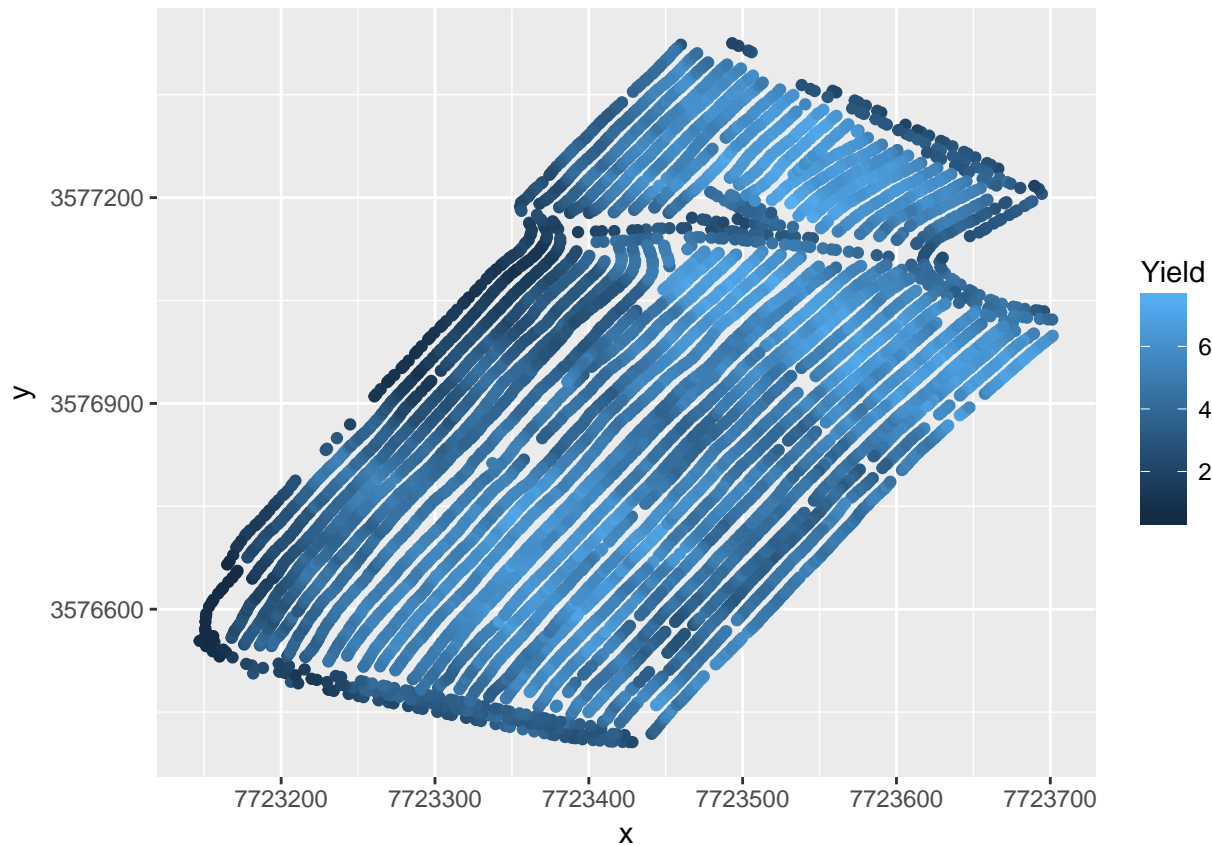
```
boundary0$y=c(3576538,3576438,3576432,3576396,3576386,3576388,3576776,3577000,3577054,3577104,3577175,
3577389,3577388,3577455,3577445,3577211,3577151,3576860,3576697,3576547)
```

```
boundary0$x[length(boundary0$x)]=boundary0$x[1]
boundary0$y[length(boundary0$y)]=boundary0$y[1]
```

```
boundary1<-cbind.data.frame(boundary0$x,boundary0$y)
colnames(boundary1)<-c("x", "y")
```

Plot data

```
ggplot(data=yield,aes(x=x,y=y,colour=Yield)) + geom_point() #+
```

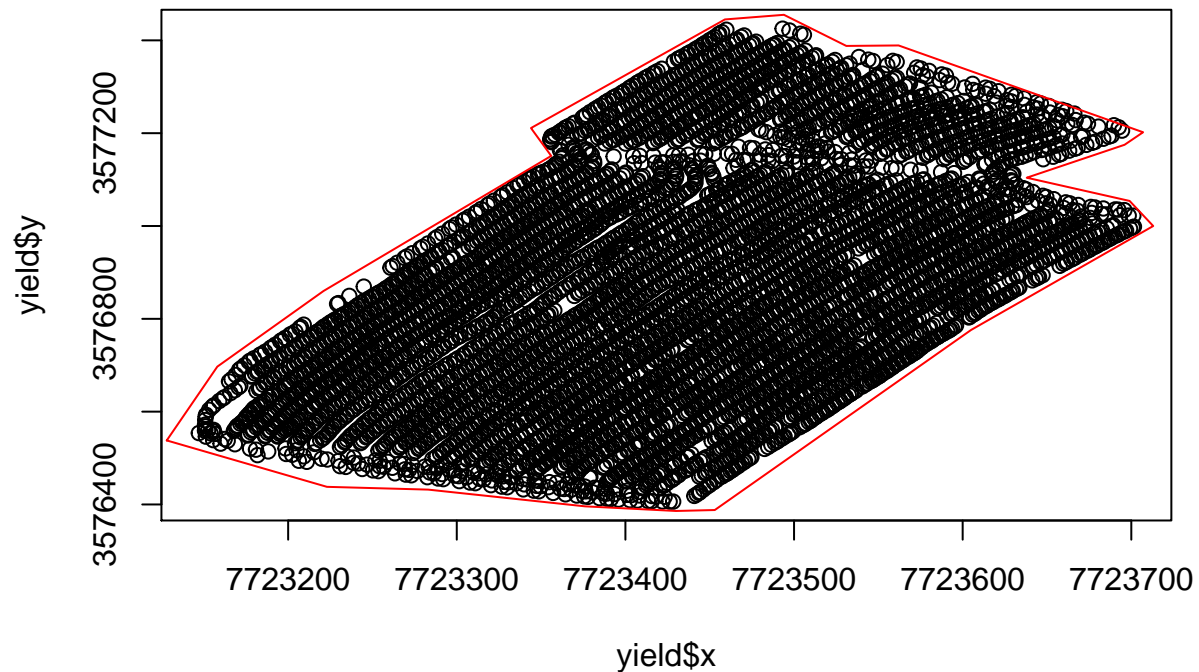


```
geom_line(data=boundary1, aes(x=x,y=y),col="red")
```

```
## mapping: x = x, y = y
## geom_line: na.rm = FALSE
## stat_identity: na.rm = FALSE
## position_identity
```

Plot data with boundary

```
plot(x=yield$x, y=yield$y)
lines(boundary0$x,boundary0$y,col="red")
```

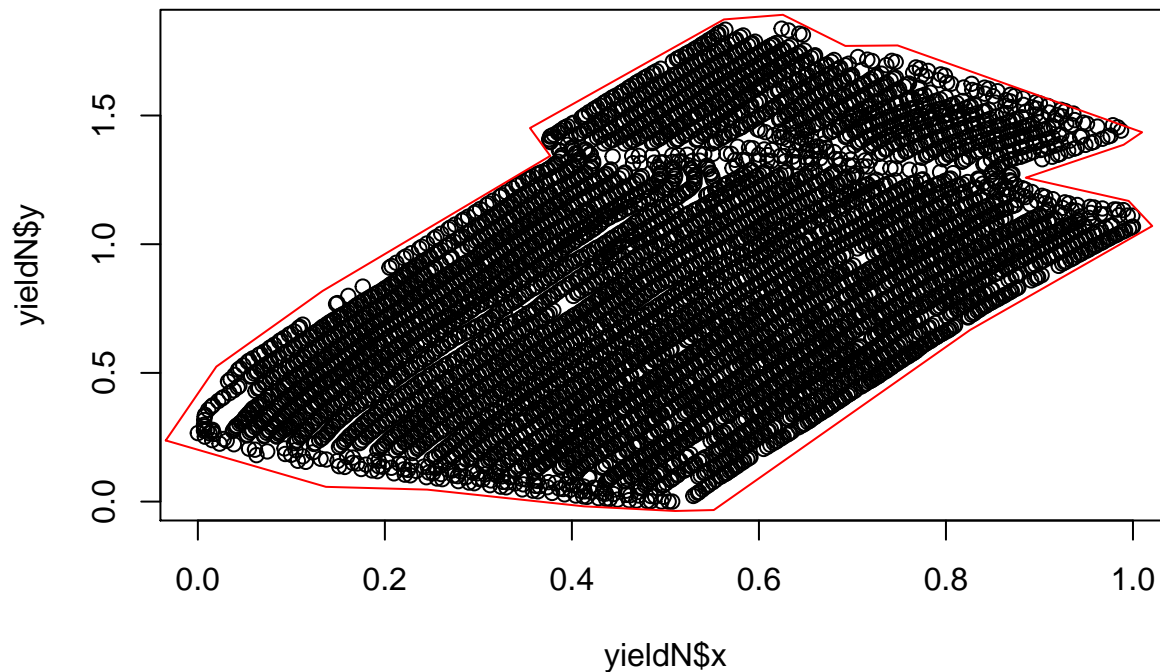


Normalize data coordinates and border in a rectangular field. x range will be equal to 1 and y range respecting the initial y/x ratio

```
resNorm = datanormX(yield, boundary0)
if (is.null(resNorm)) print("error in coordinates")
yieldN = resNorm$dataN
boundaryN = resNorm$boundaryN
xmin=resNorm$xmin
xmax=resNorm$xmax
ymin=resNorm$ymin
ymax=resNorm$ymax
ratio=resNorm$ratio
```

Plot normalized data

```
plot(x=yieldN$x, y=yieldN$y)
lines(boundaryN$x,boundaryN$y,col="red")
```



map data based on real yield data.

```
map=genMap(yieldN,seed=0,boundary=boundaryN,disp=0,nPointsK=7000)
```

```
## [1] "reading DataObj,nrow(DataObj)= 6415 ,ncol(DataObj)= 3"
## [inverse distance weighted interpolation]
```

```
boundary=map$boundary
```

Transform minSize as a percentage of frame

```
“r minSize = 0.012 # default zone surface threshold minSizeNG= 1e-3 # default threshold for both no grow
and zone grow
```

```
resT=normSize(boundaryN,minSize,minSizeNG) “
```

```
## [1] "after standardization minSize= 0.0104897399047716 minSizeNG= 0.000874144992064298"
```

```
r minSize=resT$minSize minSizeNG=resT$minSizeNG
```

## Import shape1 object

```
shape1<-geozoning::shape1
#obtention of coords
p = shape1@polygons
sp=list()
for (k in 1:length(p))
{
  sp[[k]] = (p[[k]]@Polygons)[[1]]
  co=coordinates(sp[[k]])
  co[,1]=(co[,1]-xmin)/(xmax-xmin)
  co[,2]=(co[,2]-ymin)/(xmax-xmin)
  sp[[k]]@coords=co
  sp[[k]] = polyToSp2(sp[[k]])
}
```

```

}

NZ=length(sp)
for (iZ in 1:NZ)
{
    sp=setId(sp,iZ,iZ)
}

```

calNei removes zones with n=0 or n=1 pt

```

K =calNei(sp,map$krigData,map$krigSurfVoronoi,map$krigN,simplitol=simplitol)
# zoning Z is obtained as follows
Z =K$zonePolygone
nZ=length(Z)

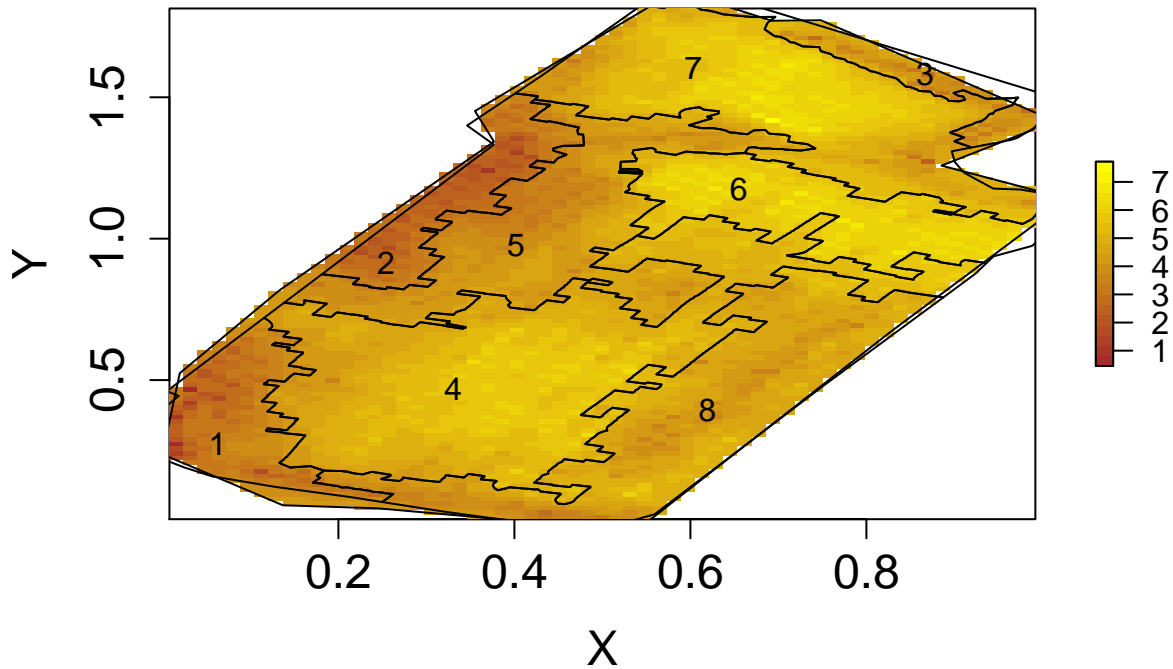
```

View external zoning

```

dispZ(map$step,map$krigGrid,zonePolygone=Z,K=K,boundary=map$boundary,nbLvl=0)

```



## NULL

Compute distance matrix between zones

```

resD = calDistance(typedist=1,map$krigData,K$listZonePoint,K$zoneN,map$krigSurfVoronoi,K$meanZone,pEr

```

Print normalized distance matrix (1 on diagonal)

```

print(normDistMat(resD$matDistanceCorr,2)) # zone pairs (6,7) and (4,8) have low distance values

```

```

##      [,1]      [,2]      [,3]      [,4]      [,5]      [,6]      [,7]
## [1,] 1.000000 0.000000 0.000000 6.703782 0.000000 0.000000 0.000000
## [2,] 0.000000 1.000000 0.000000 0.000000 4.404113 0.000000 9.876207
## [3,] 0.000000 0.000000 1.000000 0.000000 0.000000 0.000000 2.619206
## [4,] 6.703782 0.000000 0.000000 1.000000 3.131047 2.884662 0.000000
## [5,] 0.000000 4.404113 0.000000 3.131047 1.000000 10.310303 3.130948
## [6,] 0.000000 0.000000 0.000000 2.884662 10.310303 1.000000 1.654262

```

```
## [7,] 0.000000 9.876207 2.619206 0.000000 3.130948 1.654262 1.000000
## [8,] 3.737016 0.000000 0.000000 1.804517 0.000000 0.000000 0.000000
##      [,8]
## [1,] 3.737016
## [2,] 0.000000
## [3,] 0.000000
## [4,] 1.804517
## [5,] 0.000000
## [6,] 0.000000
## [7,] 0.000000
## [8,] 1.000000
```

Now compute criterion corresponding to zoning

```
crit = calCrit(resD$matDistanceCorr,K$zoneNModif,2) # crit is the mimimum distance value (between zon
print(crit)
```

```
## [1] 1.654262
```

## Session informations

```
## R version 3.4.0 (2017-04-21)
## Platform: x86_64-pc-linux-gnu (64-bit)
## Running under: Debian GNU/Linux 8 (jessie)
##
## Matrix products: default
## BLAS: /usr/lib/libblas/libblas.so.3.0
## LAPACK: /usr/lib/lapack/liblapack.so.3.0
##
## locale:
##  [1] LC_CTYPE=fr_FR.utf8      LC_NUMERIC=C
##  [3] LC_TIME=fr_FR.utf8      LC_COLLATE=fr_FR.utf8
##  [5] LC_MONETARY=fr_FR.utf8  LC_MESSAGES=fr_FR.utf8
##  [7] LC_PAPER=fr_FR.utf8     LC_NAME=C
##  [9] LC_ADDRESS=C            LC_TELEPHONE=C
## [11] LC_MEASUREMENT=fr_FR.utf8 LC_IDENTIFICATION=C
##
## attached base packages:
## [1] stats      graphics  grDevices  utils      datasets  methods    base
##
## other attached packages:
## [1] sp_1.2-4      ggplot2_2.2.1  geozoning_1.0.0 rmarkdown_1.6
##
## loaded via a namespace (and not attached):
##  [1] Rcpp_0.12.11      compiler_3.4.0
##  [3] plyr_1.8.4        tools_3.4.0
##  [5] xts_0.9-7         digest_0.6.12
##  [7] gstat_1.1-5       evaluate_0.10.1
##  [9] tibble_1.3.1      gtable_0.2.0
## [11] lattice_0.20-35   rlang_0.1.1
## [13] yaml_2.1.14       spam_1.4-0
## [15] stringr_1.2.0     knitr_1.17
## [17] raster_2.5-8      rgeos_0.3-23
## [19] RandomFieldsUtils_0.3.25 fields_8.15
```

## [21] maps_3.1.1	rprojroot_1.2
## [23] grid_3.4.0	spacetime_1.2-0
## [25] foreign_0.8-68	deldir_0.1-14
## [27] magrittr_1.5	backports_1.1.0
## [29] scales_0.4.1	htmltools_0.3.6
## [31] intervals_0.15.1	RandomFields_3.1.50
## [33] maptools_0.9-2	colorspace_1.3-2
## [35] labeling_0.3	stringi_1.1.5
## [37] lazyeval_0.2.0	munsell_0.4.3
## [39] FNN_1.1	zoo_1.8-0