Zoning real yield data

B. Charnomordic

2017-09-12

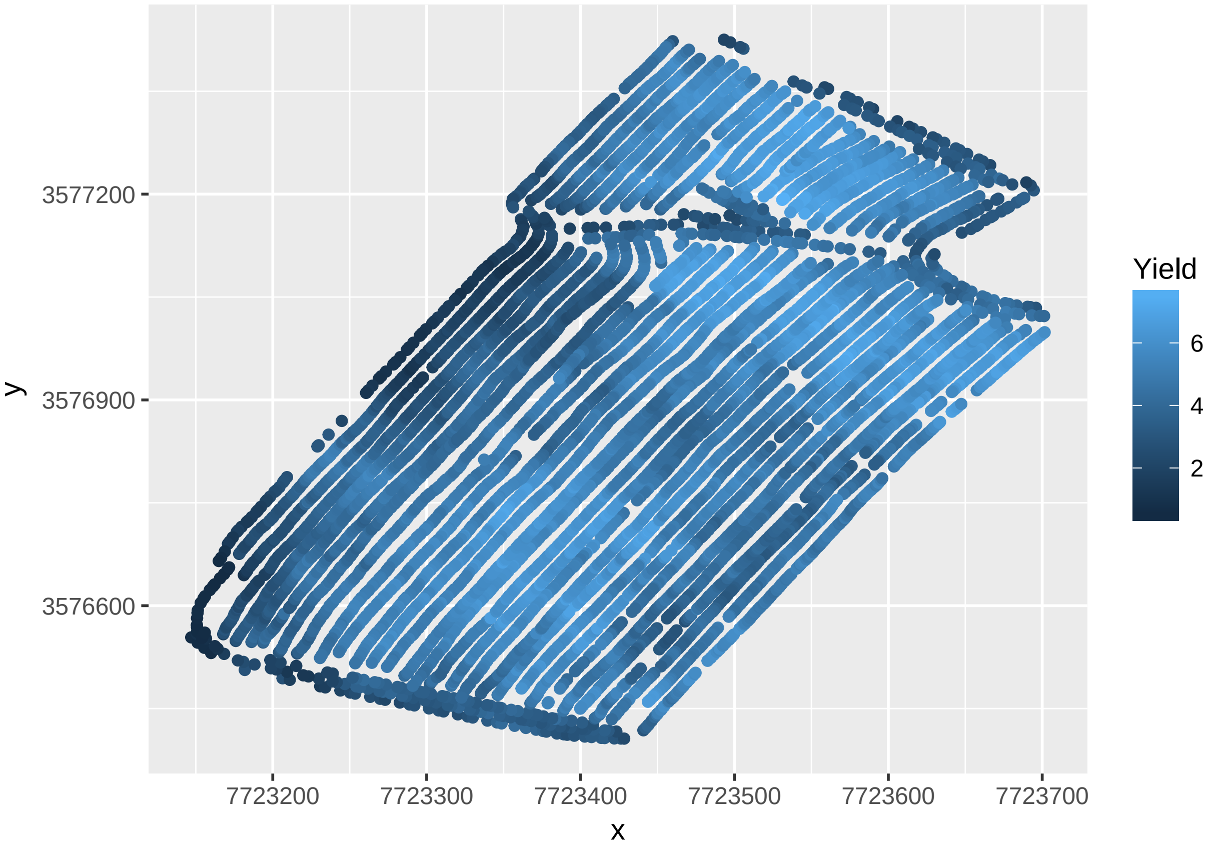
```
  library(geozoning)
  library(ggplot2)
```

This vignette illustrates the zoning with correction procedure on real yield data. It uses the yield data available in the geozoning package, where coordinates are not normalized between 0 and 1. The field is rectangular. A boundary is manually defined around the field. Then the x coordinate is normalized between 0 and 1, and the y coordinate is normalized with the ratio used for x.

Exploratory loops for size 1 to 4 probability vectors (producing 2 to 5 labels maps) are run to rank the zonings, for each number of labels independently, according to their quality criterion. Probability vectors corresponding to the best zonings for each number of labels are extracted from the loop results. Correction procedures are then run for each of these best zonings. For readability, zones are reordered by increasing average value. An example of plot is given for the best 4-label zoning, to view zones and the distribution of values within zones.

```
# Import yield data
  data(yield)
```

```
  # draw manual boundary
  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() #+
```



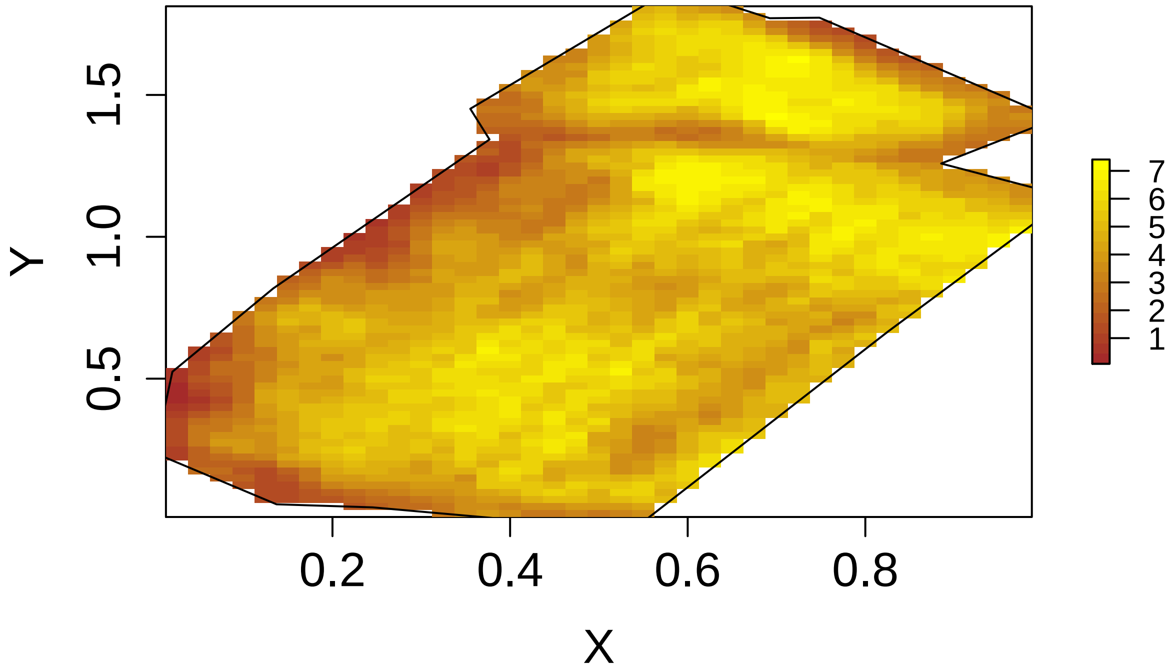
```
  geom_line(data=boundary1, aes(x=x,y=y),col="red")
```

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

```
#  x and y coordinates are normalized in genMap with (xmax-xmin) ratio

# build map data based on yield data
map=genMap(yield,seed=0,boundary=boundary0,disp=0,nPointsK=3000,krig=1)
```

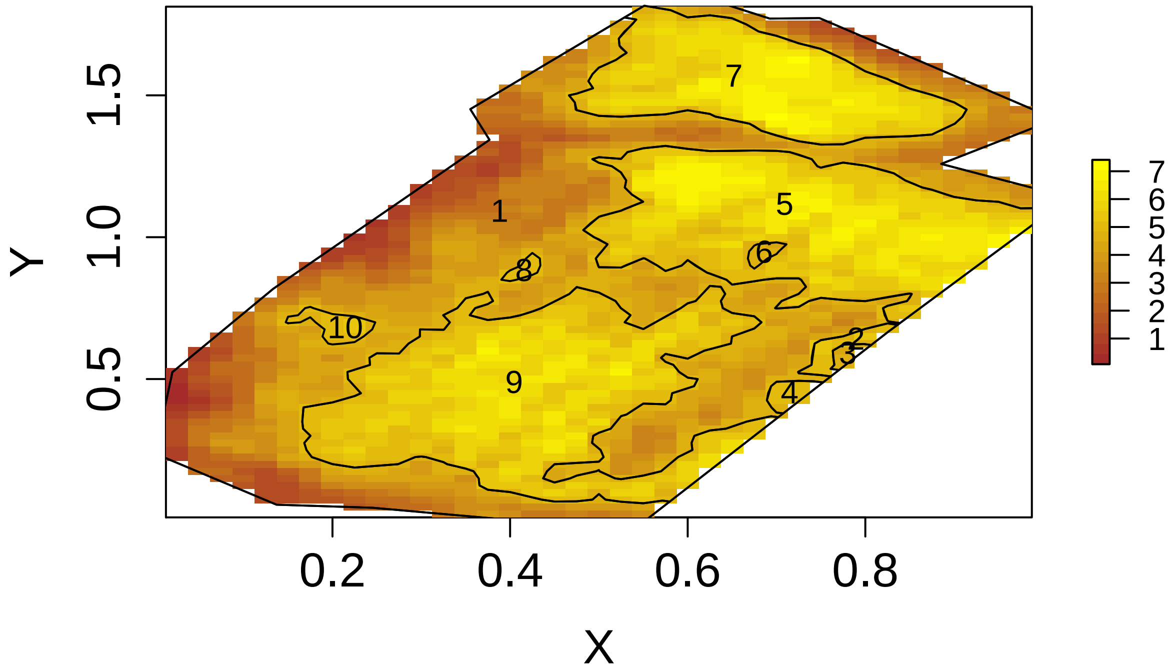
```
## [1] "reading DataObj,nrow(DataObj)= 6415 ,ncol(DataObj)= 3"
## [using ordinary kriging]
```

```
boundaryN=map$boundary
dispZ(map$step,matVal=map$krigGrid,zonePolygone=NULL,boundary=map$boundary)
```



```
## NULL
```

```
# test zoning with one quantile (median value)
qProb=0.5
ZK = initialZoning(qProb,map)
K=ZK$resZ
Z=K$zonePolygone
dispZ(map$step,map$krigGrid,zonePolygone=Z,K=K,boundary=map$boundary,nbLvl=0,id=FALSE)
```

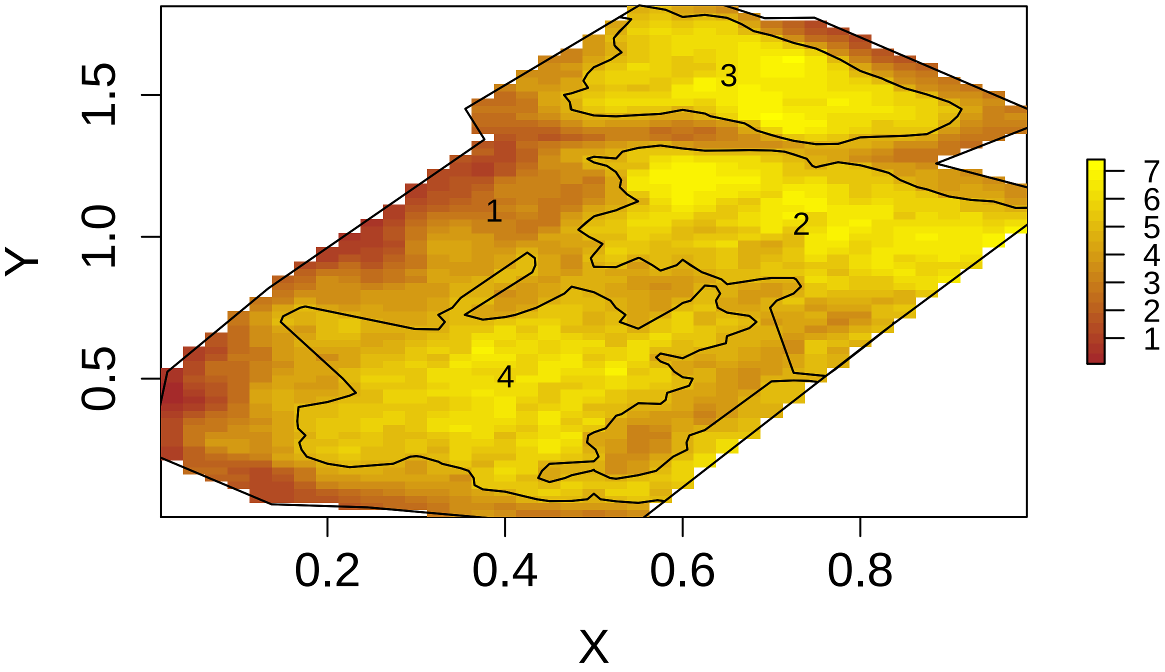


```
## NULL
```

```r
#apply corrections and display detailed information at each step
criti= correctionTree(qProb=c(0.5),map,minSize=minSize,minSizeNG=minSizeNG,disp=1)
```

```
## [1] "qProb= 0.5"
## [1] "10 zones, 6  small zones:"
## [1] "8,6,3,4,2,10"
## [1] "level=1, initial crit= 3.173"
##
## [1] "in loop level= 2 ,zone to handle initial number (id)=  8 , 1 branch(es) to examine "
## [1] "iter= 1  new zone number= 8"
## [1] "merging zone 8  with main zone 1"
## [1] "9  polygons after zone merging"
## [1] "trying to grow zone id 8 - new number 8"
## [1] "growing non isolated zone  8 (close to zone 9 )"
## [1] "junction of non isolated zone:  8 and zone 9"
## [1] "9  zones after zone growing"
##
## [1] "in loop level= 3 ,zone to handle initial number (id)=  6 , 2 branch(es) to examine "
## [1] "iter= 1  new zone number= 6"
## [1] "merging zone 6  with main zone 5"
## [1] "8  polygons after zone merging"
## [1] "trying to grow zone id 6 - new number 6"
## [1] "0  zones after zone growing"
## [1] "iter= 2  new zone number= 6"
## [1] "merging zone 6  with main zone 5"
## [1] "8  polygons after zone merging"
## [1] "trying to grow zone id 6 - new number 6"
## [1] "0  zones after zone growing"
##
## [1] "in loop level= 3 ,zone to handle initial number (id)=  3 , 2 branch(es) to examine "
## [1] "iter= 1  new zone number= 3"
## [1] "merging zone 3  with main zone 2"
## [1] "7  polygons after zone merging"
## [1] "trying to grow zone id 3 - new number 3"
## [1] "0  zones after zone growing"
## [1] "iter= 2  new zone number= 3"
## [1] "merging zone 3  with main zone 2"
## [1] "7  polygons after zone merging"
## [1] "trying to grow zone id 3 - new number 3"
## [1] "0  zones after zone growing"
##
## [1] "in loop level= 3 ,zone to handle initial number (id)=  4 , 2 branch(es) to examine "
## [1] "iter= 1  new zone number= 3"
## [1] "merging zone 3  with main zone 1"
## [1] "6  polygons after zone merging"
## [1] "trying to grow zone id 4 - new number 3"
## [1] "growing non isolated zone  4 (close to zone 9 )"
## [1] "junction of non isolated zone:  3 and zone 6"
## [1] "6  zones after zone growing"
## [1] "iter= 2  new zone number= 3"
## [1] "merging zone 3  with main zone 1"
## [1] "6  polygons after zone merging"
## [1] "trying to grow zone id 4 - new number 3"
## [1] "growing non isolated zone  4 (close to zone 9 )"
## [1] "junction of non isolated zone:  3 and zone 6"
## [1] "6  zones after zone growing"
##
## [1] "in loop level= 3 ,zone to handle initial number (id)=  2 , 4 branch(es) to examine "
## [1] "iter= 1  new zone number= 2"
## [1] "merging zone 2  with main zone 1"
## [1] "5  polygons after zone merging"
## [1] "trying to grow zone id 2 - new number 2"
## [1] "growing non isolated zone  2 (close to zone 5 )"
## [1] "junction of non isolated zone:  2 and zone 3"
## [1] "5  zones after zone growing"
## [1] "iter= 2  new zone number= 2"
## [1] "merging zone 2  with main zone 1"
## [1] "5  polygons after zone merging"
## [1] "trying to grow zone id 2 - new number 2"
## [1] "growing non isolated zone  2 (close to zone 5 )"
## [1] "junction of non isolated zone:  2 and zone 3"
## [1] "5  zones after zone growing"
## [1] "iter= 3  new zone number= 2"
## [1] "merging zone 2  with main zone 1"
## [1] "5  polygons after zone merging"
## [1] "trying to grow zone id 2 - new number 2"
## [1] "growing non isolated zone  2 (close to zone 5 )"
## [1] "junction of non isolated zone:  2 and zone 3"
## [1] "5  zones after zone growing"
## [1] "iter= 4  new zone number= 2"
## [1] "merging zone 2  with main zone 1"
## [1] "5  polygons after zone merging"
## [1] "trying to grow zone id 2 - new number 2"
## [1] "growing non isolated zone  2 (close to zone 5 )"
## [1] "junction of non isolated zone:  2 and zone 3"
## [1] "5  zones after zone growing"
##
## [1] "in loop level= 3 ,zone to handle initial number (id)=  10 , 8 branch(es) to examine "
## [1] "iter= 1  new zone number= 5"
## [1] "merging zone 5  with main zone 1"
## [1] "4  polygons after zone merging"
## [1] "trying to grow zone id 10 - new number 5"
## [1] "growing non isolated zone  10 (close to zone 9 )"
## [1] "junction of non isolated zone:  5 and zone 4"
## [1] "4  zones after zone growing"
## [1] "iter= 2  new zone number= 5"
## [1] "merging zone 5  with main zone 1"
## [1] "4  polygons after zone merging"
## [1] "trying to grow zone id 10 - new number 5"
## [1] "growing non isolated zone  10 (close to zone 9 )"
## [1] "junction of non isolated zone:  5 and zone 4"
## [1] "4  zones after zone growing"
## [1] "iter= 3  new zone number= 5"
## [1] "merging zone 5  with main zone 1"
## [1] "4  polygons after zone merging"
## [1] "trying to grow zone id 10 - new number 5"
## [1] "growing non isolated zone  10 (close to zone 9 )"
## [1] "junction of non isolated zone:  5 and zone 4"
## [1] "4  zones after zone growing"
## [1] "iter= 4  new zone number= 5"
## [1] "merging zone 5  with main zone 1"
## [1] "4  polygons after zone merging"
## [1] "trying to grow zone id 10 - new number 5"
## [1] "growing non isolated zone  10 (close to zone 9 )"
## [1] "junction of non isolated zone:  5 and zone 4"
## [1] "4  zones after zone growing"
## [1] "iter= 5  new zone number= 5"
## [1] "merging zone 5  with main zone 1"
## [1] "4  polygons after zone merging"
## [1] "trying to grow zone id 10 - new number 5"
## [1] "growing non isolated zone  10 (close to zone 9 )"
## [1] "junction of non isolated zone:  5 and zone 4"
## [1] "4  zones after zone growing"
## [1] "iter= 6  new zone number= 5"
## [1] "merging zone 5  with main zone 1"
## [1] "4  polygons after zone merging"
## [1] "trying to grow zone id 10 - new number 5"
## [1] "growing non isolated zone  10 (close to zone 9 )"
## [1] "junction of non isolated zone:  5 and zone 4"
## [1] "4  zones after zone growing"
## [1] "iter= 7  new zone number= 5"
## [1] "merging zone 5  with main zone 1"
## [1] "4  polygons after zone merging"
## [1] "trying to grow zone id 10 - new number 5"
## [1] "growing non isolated zone  10 (close to zone 9 )"
## [1] "junction of non isolated zone:  5 and zone 4"
## [1] "4  zones after zone growing"
## [1] "iter= 8  new zone number= 5"
## [1] "merging zone 5  with main zone 1"
## [1] "4  polygons after zone merging"
## [1] "trying to grow zone id 10 - new number 5"
## [1] "growing non isolated zone  10 (close to zone 9 )"
## [1] "junction of non isolated zone:  5 and zone 4"
## [1] "4  zones after zone growing"
## [1] "16 zonings in last level"
## [1] "length(crit[[ 1 ]])= 1"
## [1] 3.17308
## [1] "length(crit[[ 2 ]])= 16"
##  [1] 4.043206 3.879349 3.802320 3.943529 4.014831 3.883418 3.873248
##  [8] 4.025509 4.028444 3.882210 3.840547 3.987632 4.007380 3.892006
## [15] 3.914194 4.058766
```

```r
# search for best criterion at last level (once all corrections have been applied)
res=searchNODcrit1(qProb,criti)
b=res$ind[[1]][1]
K=criti$zk[[2]][[b]]
Z=K$zonePolygone
dispZ(map$step,map$krigGrid,zonePolygone=Z,K=K,boundary=map$boundary,nbLvl=0,id=FALSE)
```



```
## NULL
```

```
#
#Run exploratory loops to find the best zonings for nL=2 to 5
#Set reasonable values for minimum size of zones and size threshold for not trying to grow zones (just
minSizeNG=1e-2
mimSize=2e-2
#
ro1=loopQ1(map,disp=0,step=0.15,minSize=minSize,minSizeNG=minSizeNG)
```

```
## [1] "0.05 criterion= 7.799 cost= 1.499 costL= 1.499 nz= 3"
## [1] "0.2 criterion= 6.056 cost= 0.869 costL= 0.869 nz= 3"
## [1] "0.35 criterion= 5.208 cost= 0.655 costL= 0.679 nz= 4"
## [1] "0.5 criterion= 4.043 cost= 0.753 costL= 0.766 nz= 4"
## [1] "0.65 criterion= 3.547 cost= 0.916 costL= 0.923 nz= 5"
## [1] "0.8 criterion= 3.519 cost= 1.263 costL= 1.265 nz= 4"
## [1] "0.95 criterion= 3.687 cost= 1.855 costL= 1.855 nz= 3"
```

```
ro2=loopQ2(map,disp=0,step=0.15,minSize=minSize,minSizeNG=minSizeNG)
```

```
## [1] "0.05 0.275 criterion= 5.639 cost= 0.639 costL= 0.64 nz= 5 nq= 2"
## [1] "0.05 0.425 criterion= 4.912 cost= 0.463 costL= 0.485 nz= 7 nq= 2"
## [1] "0.05 0.575 criterion= 4.455 cost= 0.537 costL= 0.548 nz= 6 nq= 2"
## [1] "0.05 0.725 criterion= 3.943 cost= 0.73 costL= 0.736 nz= 6 nq= 2"
## [1] "0.05 0.875 criterion= 3.885 cost= 1.121 costL= 1.122 nz= 5 nq= 2"
## [1] "0.2 0.425 criterion= 4.816 cost= 0.396 costL= 0.403 nz= 8 nq= 2"
## [1] "0.2 0.575 criterion= 5.501 cost= 0.307 costL= 0.325 nz= 7 nq= 2"
## [1] "0.2 0.725 criterion= 4.819 cost= 0.363 costL= 0.376 nz= 7 nq= 2"
## [1] "0.2 0.875 criterion= 4.561 cost= 0.571 costL= 0.578 nz= 6 nq= 2"
## [1] "0.35 0.575 criterion= 3.943 cost= 0.407 costL= 0.442 nz= 7 nq= 2"
## [1] "0.35 0.725 criterion= 4.553 cost= 0.396 costL= 0.426 nz= 7 nq= 2"
## [1] "0.35 0.875 criterion= 4.74 cost= 0.501 costL= 0.525 nz= 6 nq= 2"
## [1] "0.5 0.725 criterion= 3.295 cost= 0.652 costL= 0.658 nz= 7 nq= 2"
## [1] "0.5 0.875 criterion= 3.785 cost= 0.681 costL= 0.682 nz= 6 nq= 2"
## [1] "0.65 0.875 criterion= 3.547 cost= 0.881 costL= 0.882 nz= 7 nq= 2"
```

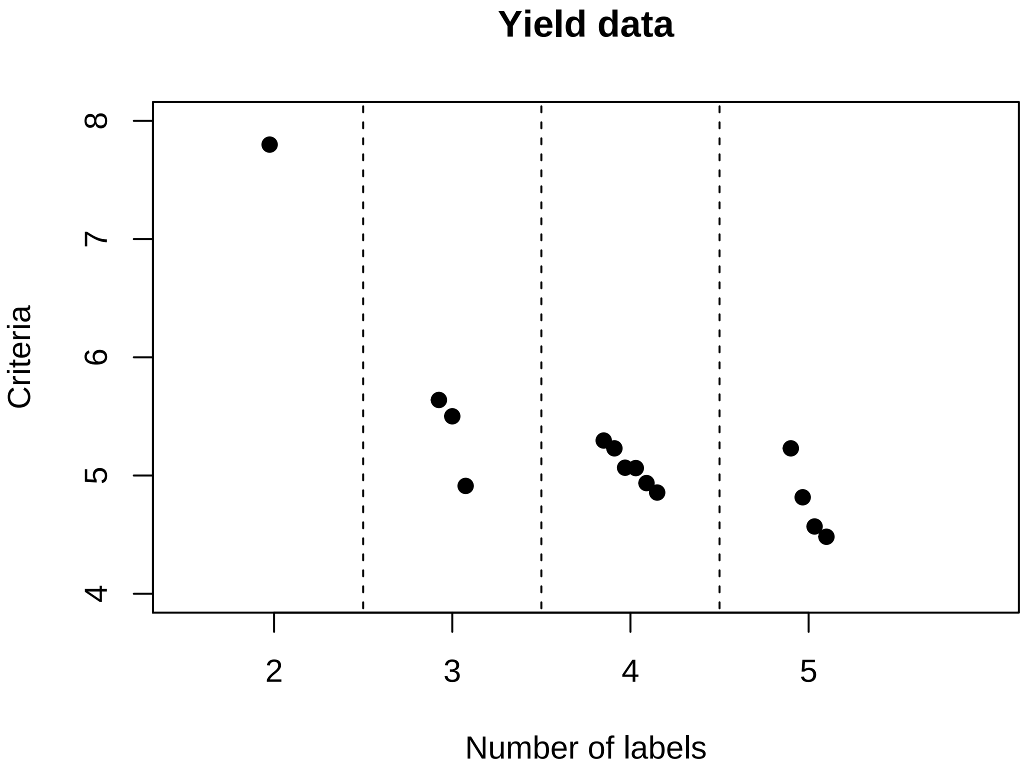
```
ro3=loopQ3(map,disp=0,step=0.15,minSize=minSize,minSizeNG=minSizeNG)
```

```
## [1] "0.05 0.275 0.5 criterion= 4.482 cost= 0.283 costL= 0.297 nz= 8 nq= 3"
## [1] "0.05 0.275 0.65 criterion= 5.23 cost= 0.243 costL= 0.251 nz= 9 nq= 3"
## [1] "0.05 0.275 0.8 criterion= 4.936 cost= 0.318 costL= 0.321 nz= 8 nq= 3"
## [1] "0.05 0.275 0.95 criterion= 4.856 cost= 0.56 costL= 0.561 nz= 7 nq= 3"
## [1] "0.05 0.425 0.65 criterion= 4.569 cost= 0.283 costL= 0.312 nz= 11 nq= 3"
## [1] "0.05 0.425 0.8 criterion= 5.296 cost= 0.285 costL= 0.31 nz= 10 nq= 3"
## [1] "0.05 0.425 0.95 criterion= 5.066 cost= 0.413 costL= 0.435 nz= 9 nq= 3"
## [1] "0.05 0.575 0.8 criterion= 3.905 cost= 0.468 costL= 0.472 nz= 9 nq= 3"
## [1] "0.05 0.575 0.95 criterion= 3.698 cost= 0.515 costL= 0.52 nz= 8 nq= 3"
## [1] "0.05 0.725 0.95 criterion= 3.382 cost= 0.719 costL= 0.722 nz= 8 nq= 3"
## [1] "0.2 0.425 0.65 criterion= 4.816 cost= 0.215 costL= 0.23 nz= 12 nq= 3"
## [1] "0.2 0.425 0.8 criterion= 4.816 cost= 0.218 costL= 0.228 nz= 11 nq= 3"
## [1] "0.2 0.425 0.95 criterion= 4.816 cost= 0.346 costL= 0.353 nz= 10 nq= 3"
## [1] "0.2 0.575 0.8 criterion= 5.063 cost= 0.239 costL= 0.249 nz= 10 nq= 3"
## [1] "0.2 0.575 0.95 criterion= 3.698 cost= 0.286 costL= 0.297 nz= 9 nq= 3"
## [1] "0.2 0.725 0.95 criterion= 3.382 cost= 0.352 costL= 0.362 nz= 9 nq= 3"
## [1] "0.35 0.575 0.8 criterion= 3.943 cost= 0.339 costL= 0.365 nz= 10 nq= 3"
## [1] "0.35 0.575 0.95 criterion= 3.698 cost= 0.385 costL= 0.414 nz= 9 nq= 3"
## [1] "0.35 0.725 0.95 criterion= 3.382 cost= 0.385 costL= 0.412 nz= 9 nq= 3"
## [1] "0.5 0.725 0.95 criterion= 3.295 cost= 0.641 costL= 0.644 nz= 9 nq= 3"
```

```
ro4=loopQ4(map,disp=0,step=0.15,minSize=minSize,minSizeNG=minSizeNG)
```

```
## [1] "0.05 0.275 0.5 0.725 criterion= 4.173 cost= 0.182 costL= 0.188 nz= 11 nq= 4"
## [1] "0.05 0.275 0.5 0.875 criterion= 4.482 cost= 0.211 costL= 0.213 nz= 10 nq= 4"
## [1] "0.05 0.275 0.65 0.875 criterion= 5.23 cost= 0.209 costL= 0.21 nz= 11 nq= 4"
## [1] "0.05 0.425 0.65 0.875 criterion= 4.569 cost= 0.248 costL= 0.271 nz= 13 nq= 4"
## [1] "0.2 0.425 0.65 0.875 criterion= 4.816 cost= 0.181 costL= 0.189 nz= 14 nq= 4"
```

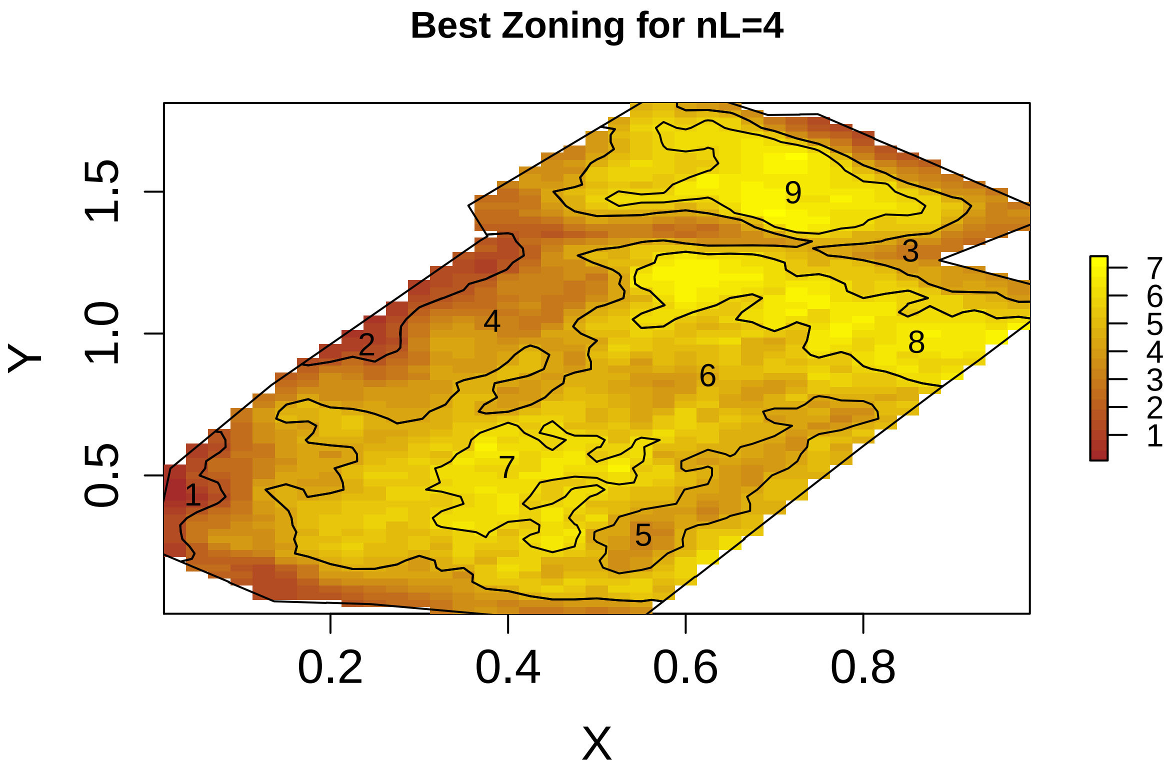
```
# study results, first plot criteria
best=figCritN(m1=ro1,m2=ro2,m3=ro3,m4=ro4,NEW=TRUE,ONE=TRUE,title="Yield data",pdf=NULL)
```



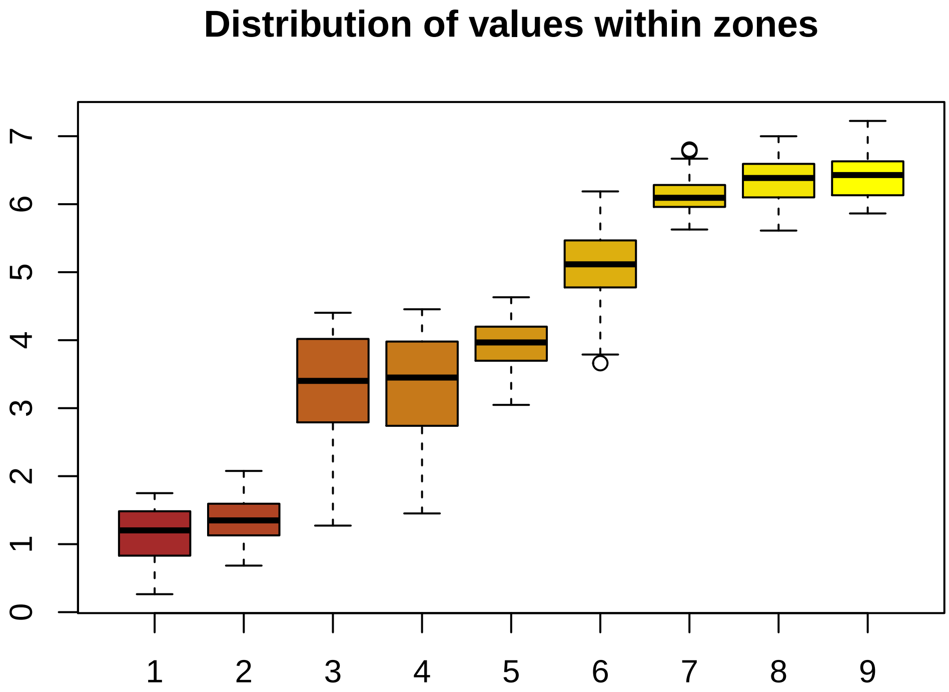
```
# best probability vectors for a number of labels between 2 and 5
m1=best[1]
m2=best[2:3]
m3=best[4:6]
m4=best[7:10]
# build best zonings for a given number of labels (between 2 and 5), that implies calculating correctio
# assign corresponding objects Z1, Z2, Z3, Z4 and values val1, val2, val3, val4
for (k in 1:4)
    {
    mk=get(paste("m",k,sep=""))
    critk=correctionTree(mk,map,disp=0,SAVE=T)
    res=searchNODcrit1(mk,critk)
    jj=res$ind[[1]][1]
    zk=critk$zk
    K=zk[[length(zk)]][[jj]]
    # order zones by increasing average zone value
    valk=valZ(map,K)$val
    ordk=valZ(map,K)$ord
    Zk=K$zonePolygone
    ii=0
    for (iZ in ordk)
            {
        ii=ii+1
            Zk=setId(Zk,iZ,ii)
             }
    Zkname=paste("Z",k,sep="")
    assign(Zkname,Zk)
    valkname=paste("val",k,sep="")
    assign(valkname,valk)
    }

# for 4 labels, display data, best zonings and distribution of values within zones (boxplot)
   palCol=colorRampPalette(c("brown","yellow"))
   dispZ(map$step,map$krigGrid,zonePolygone=Z3,id=TRUE,palCol=palCol(length(Z3)))
```

```
## NULL
```

```
   title("Best Zoning for nL=4")
```



```
   boxplot(val3,col=palCol(length(val3)))
    title("Distribution of values within zones")
```



```
# 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] grid      stats     graphics  grDevices utils     datasets  methods
## [8] base
##
## other attached packages:
## [1] fields_8.15     maps_3.1.1      spam_1.4-0      rgeos_0.3-23
## [5] 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              stringr_1.2.0
## [15] knitr_1.17               raster_2.5-8
## [17] RandomFieldsUtils_0.3.25 rprojroot_1.2
## [19] spacetime_1.2-0          foreign_0.8-68
## [21] deldir_0.1-14            magrittr_1.5
## [23] backports_1.1.0          scales_0.4.1
## [25] htmltools_0.3.6          intervals_0.15.1
## [27] RandomFields_3.1.50      maptools_0.9-2
## [29] colorspace_1.3-2         labeling_0.3
## [31] stringi_1.1.5            lazyeval_0.2.0
## [33] munsell_0.4.3            FNN_1.1
## [35] zoo_1.8-0
```