

View zoning together with zone distribution

B. Charnomordic

2017-09-12

Contents

Step 1	1
Step 2	1
Step 3	2

```
library(geozoning)
library(sp)
library(fields)
```

This vignette shows how to use the zoning with corrections procedure. It first calls the `correctionTree` function, which generates a binary tree of corected zonings and then searches for the best corrected zoning. It plots the resulting zoning as well as the distribution of values within each zone.

Step 1

A map object is simulated with a Gaussian field and a variogram model. 450 points are randomly allocated on a square field of size 1. Then 1936 points are kriged on a regular grid using inverse distance weighted interpolation. A Delaunay tessellation yields point neighborhood in the sense of Voronoi.

```
seed=40
map=genMap(DataObj=NULL,seed=seed,disp=FALSE,krig=1,Vnugget=1.2)
```

```
## [1] "DataObj=NULL, generating DataObj-seed= 40"
## [using ordinary kriging]
```

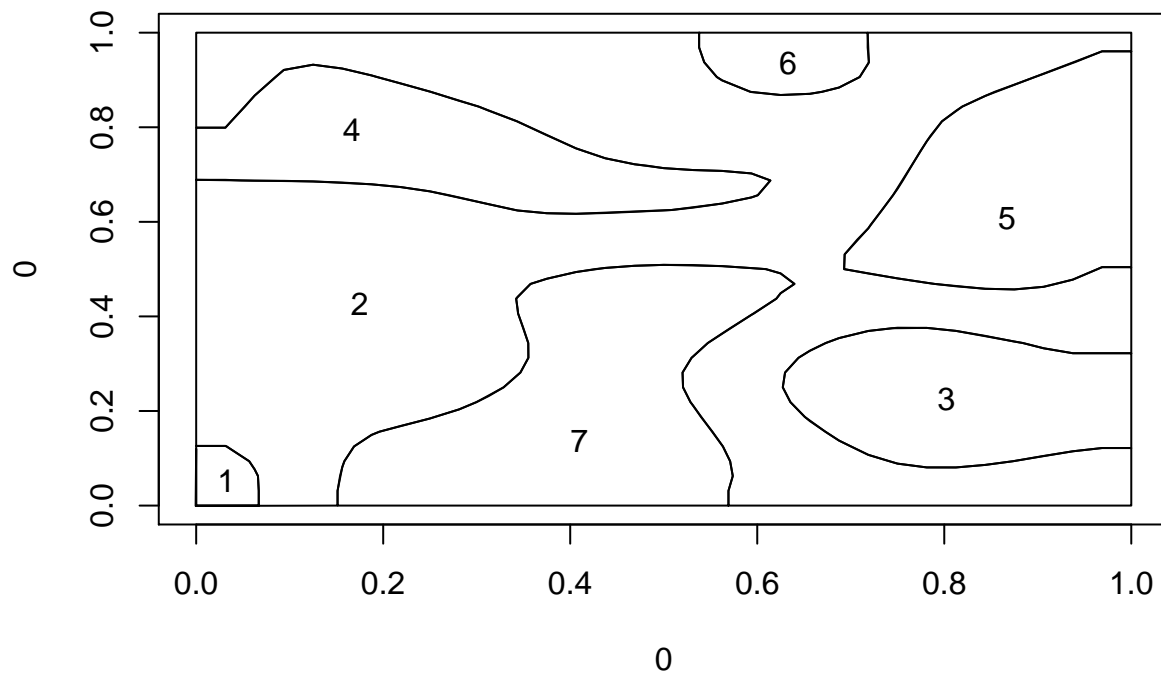
```
# Check the mean and standard deviation of generated data.
meanvarSimu(map)
```

```
##      raw mean kriged mean      raw sd   kriged sd
##      9.072091   8.858121   2.229093   1.964175
```

Step 2

Generate zoning Z from map and quantile vector with small zone correction. Small zones are kept until the end so that they get more chance to grow. `LASTPASS=FALSE` does not remove in a final step zones that are still too small.

```
#####
qProb=c(0.275,0.8)
criti=correctionTree(qProb,map,LASTPASS=FALSE)
# find best corrected zoning among last level zonings (7 zones in this case)
res=searchNODcrit1(qProb,criti)
b=res$ind[[1]][1]
K=criti$zk[[2]][[b]]
Z=K$zonePolygone
plotZ(Z,id=TRUE)
```

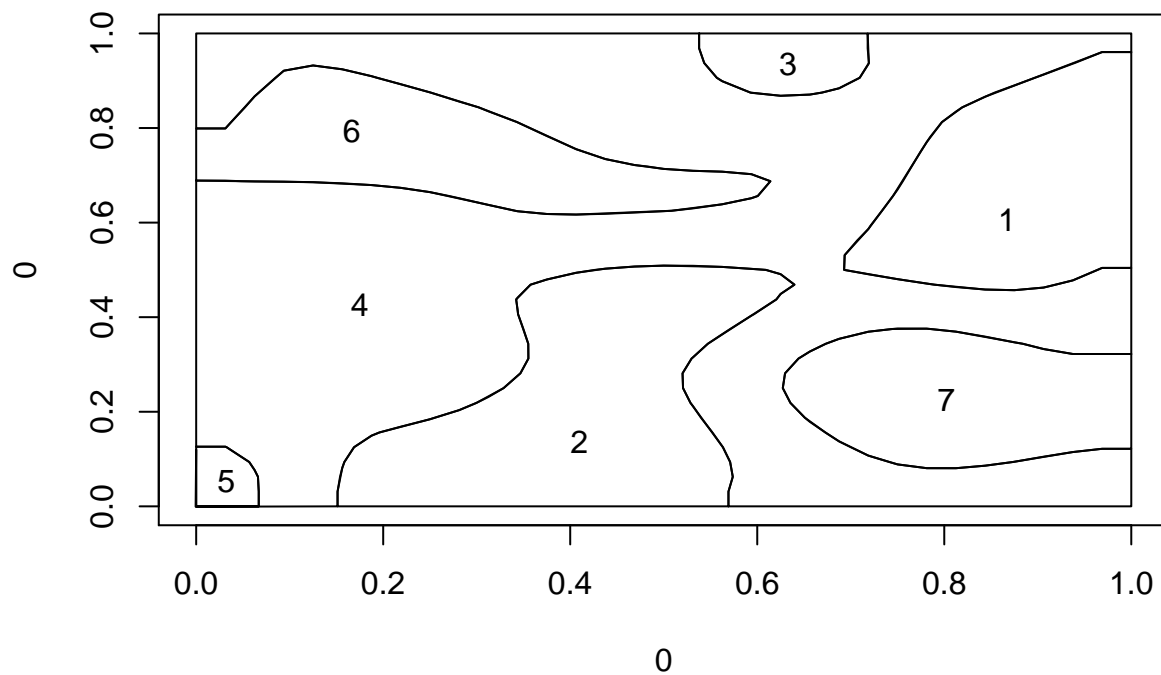


NULL

Step 3

Order zones by attribute mean value

```
val=valZ(map,K)$val
ord=valZ(map,K)$ord
Z=orderZ(Z,ord)
plotZ(Z,id=TRUE)
```

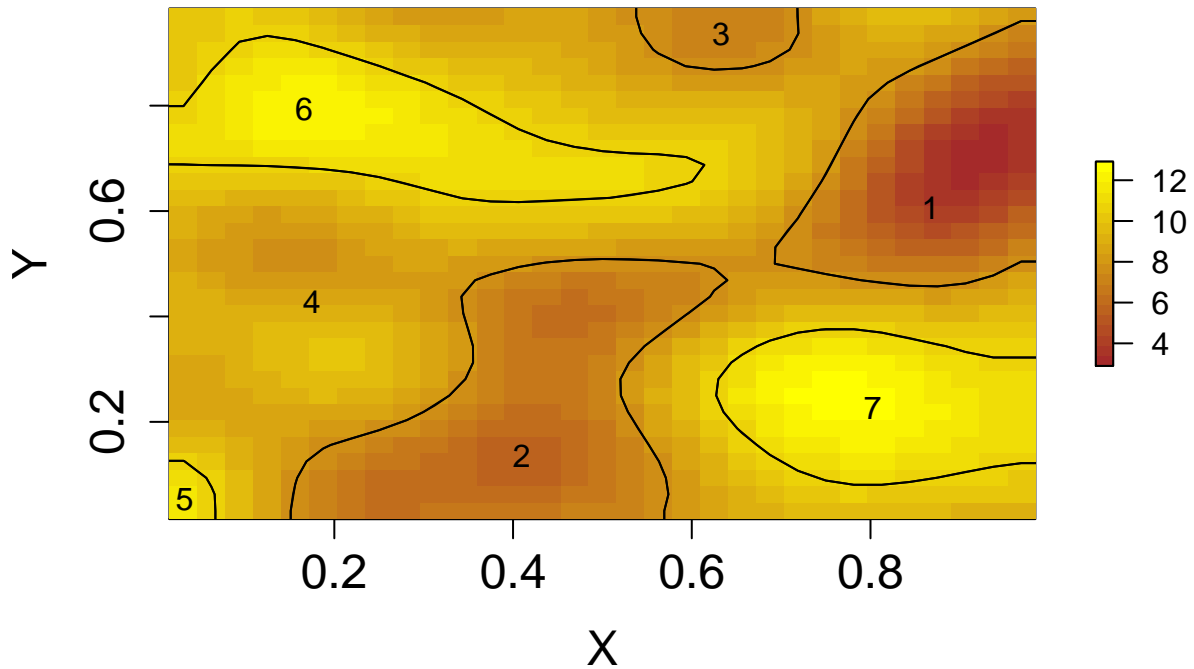


```
## NULL
```

```
#plot zoning together with zone distribution of values
```

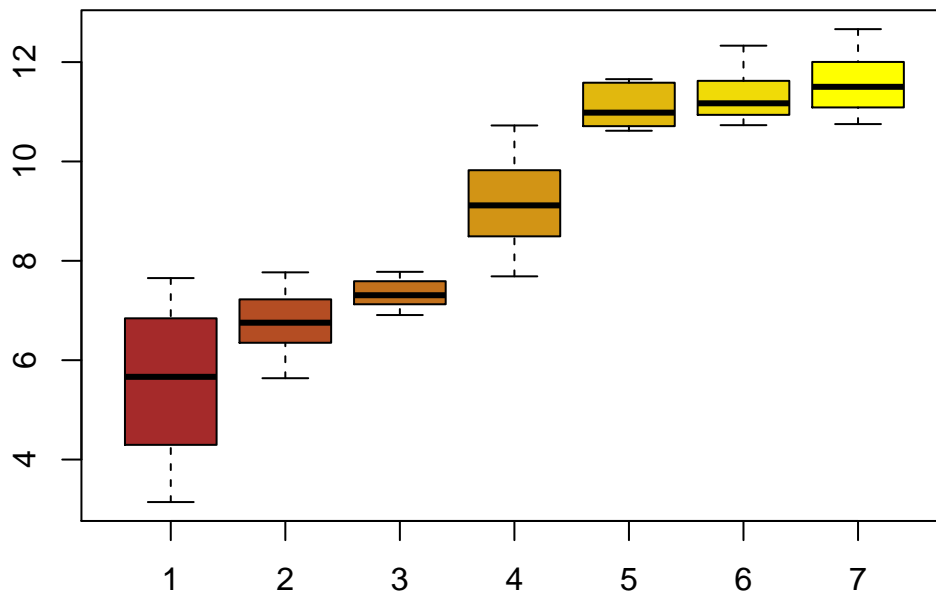
```
palCol=colorRampPalette(c("brown","yellow"))
```

```
dispZ(map$step,map$krigGrid,zonePolygone=Z,id=TRUE,palCol=palCol(length(val)))
```



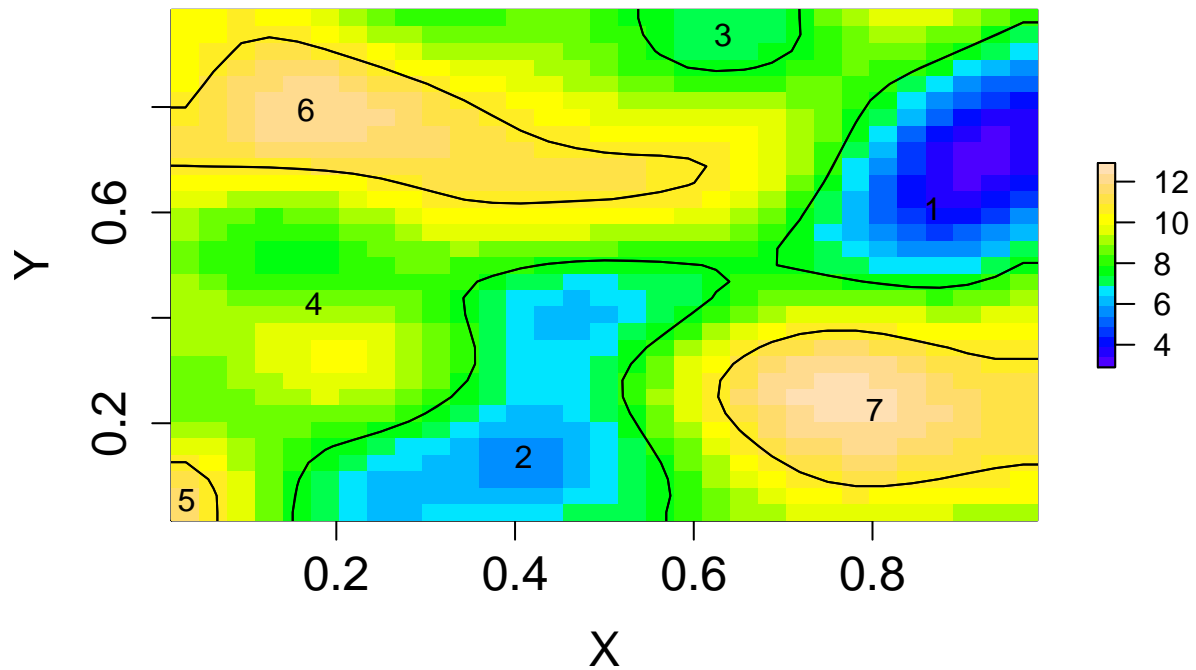
```
## NULL
```

```
boxplot(val,col=palCol(length(val)))
```



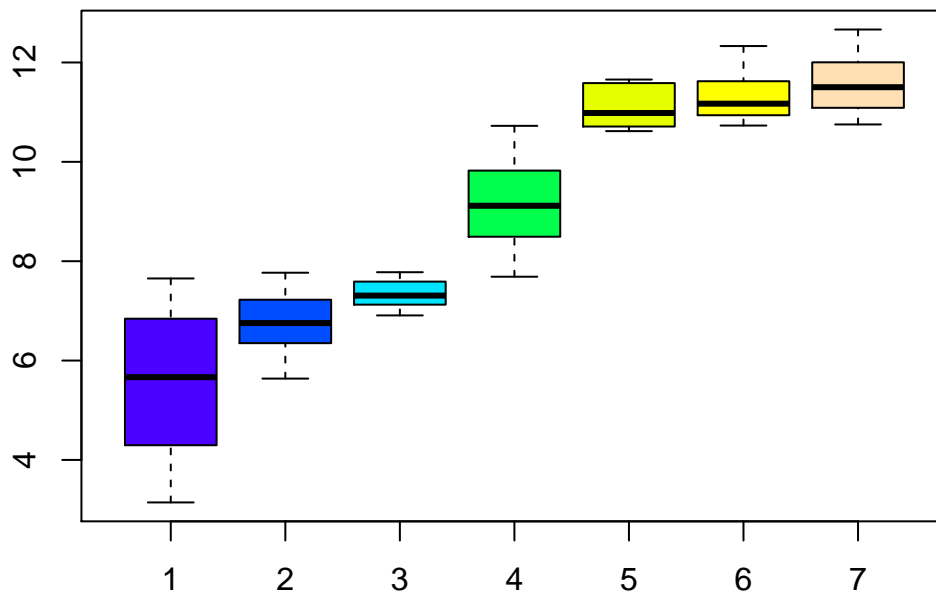
```
# another color palette
```

```
dispZ(map$step,map$krigGrid,zonePolygone=Z,id=TRUE,palCol=topo.colors)
```



```
## NULL
```

```
boxplot(val,col=topo.colors(length(val)))
```



```
# 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

```