CODE

Delimitation of Nodal Regions Based on Transport Flows

Thomas Blachier - *21 August 2019*

Clu001

Call of libraries

HIDE

```
library(dplyr)
library(ggplot2)
library(ggvoronoi)
library(tripack)
library(readr)
library(deldir)
library(SDMTools)
library(readr)
library(arsenal)
library(ggforce)
library(tidyverse)
library(mclust)
```

Importation of data from the datastore

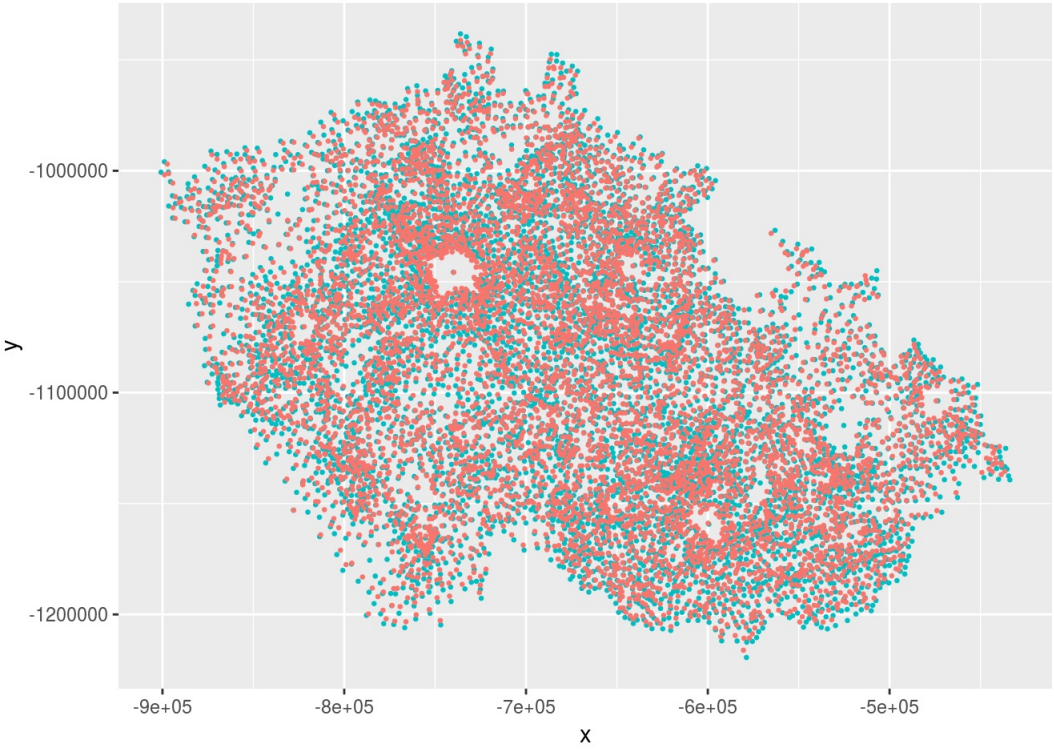
HIDE

```
df.path="~/datastore/AlfaDWbh6/ext-clu/dat"
df.files = list.files(df.path, recursive = T, full.names = T)
odm <- read_csv(df.files [grepl ("odm", df.files)])
pop <- read_csv(df.files [grepl ("pop", df.files)])
reg <- read_csv(df.files [grepl ("reg", df.files)])
df.path2 = "~/datastore/AlfaDWbh6/ext-clu/dat2"
df.files2 = list.files(df.path2, recursive = T, full.names = T)
cisreg_l2g <- read_csv(df.files2 [grepl ("cisreg_l2g", df.files2)])
cisreg_cz0 <- read_csv(df.files2 [grepl ("cisreg_cz0", df.files2)])
attrasreg_1561366112096 <- read_csv(df.files2 [grepl ("attrasreg_1561366112096", df.files2)])
cisreg_la2 <- read_csv(df.files2 [grepl ("cisreg_la2", df.files2)])
intpol_pro_s3e_vsb_sacz0_com_geomapfor_l2g_cen<- read_csv(df.files2 [grepl ("intpol_pro_s3e_vsb_sac
z0_com_geomapfor_l2g_cen", df.files2)])
intpol_pro_s3e_vsb_sacz0_com_geomapfor_l2g_pol<- read_csv(df.files2 [grepl ("intpol_pro_s3e_vsb_sac
z0_com_geomapfor_l2g_pol", df.files2)])
# Creation of set by joining odm (flow between cities) and pop (population of the cities) .
```

Creation of a dataset of city coordinates corrected by taking population and flows of population

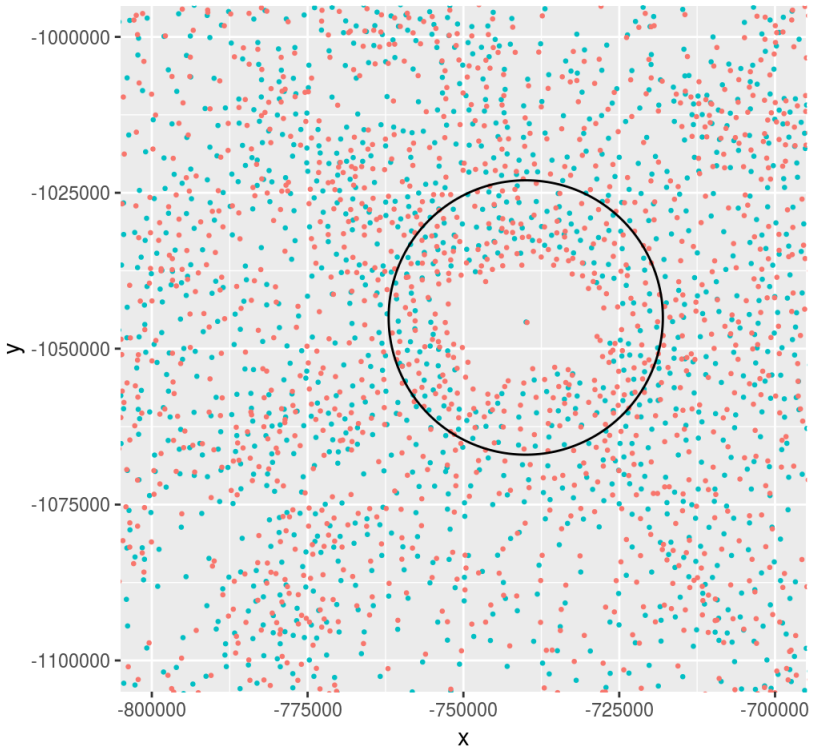
HIDE

```
set = inner_join(odm,rename(pop,"reg" = "cisdan"),by="reg")
p = reg
p$ZobZkr = NULL
p$GraOrp = NULL
p$pix = NULL
p$lon = NULL
p$lat = NULL
# We take only the coordinates of the cities and we create an alias.
set = inner_join(set,rename(p,"reg" = "cisdan"),by = "reg")
set = rename(set,"xa"="x")
set = rename(set,"ya"="y")
set = inner_join(set,rename(p,"re2" = "cisdan"),by = "re2")
set = rename(set,"xb"="x")
set = rename(set,"yb"="y")
# we join set with the coordinates of the starting city and the ending city for each flow.
set$xab = set$xb - set$xa
set$yab = set$yb - set$ya
# Creation of the vectors start to end for each flow
set = filter(set,set$d01 > 0)
set = filter(set,set$popupd_d01 > 0)
# Elimination of the null values of populations and flows .
set$x_vector_flow = (set$xab * set$d01)/set$popupd_d01
set$y_vector_flow = (set$yab * set$d01)/set$popupd_d01
# Application of the formula according to the following  model (https://docs.google.com/document/d/
1uwNQuylkccQlkgxTCjGKvB_X2BLYoQue8DZN8sRO00U/edit)
q1 = set%>%
  group_by(reg)%>%
  summarize(new_vectors_x = sum(x_vector_flow,na.rm= TRUE))
q2 = set%>%
  group_by(reg)%>%
  summarize(new_vectors_y = sum(y_vector_flow,na.rm= TRUE))
# Sum of the flows when they have the same starting city
q = inner_join(q1,q2,by = "reg")
q = inner_join(q,rename(p,"reg"="cisdan") ,by="reg")
q$x_final = q$new_vectors_x + q$x
q$y_final = q$new_vectors_y + q$y
# Still application of the formula
omega = q
omega$new_vectors_x = NULL
omega$new_vectors_y = NULL
omega$x = NULL
omega$y = NULL
omega$reg = NULL
omega = rename(omega,"x" = "x_final")
omega = rename(omega,"y" = "y_final")
omega = na.omit(omega)


U = ggplot() +geom_point(data=reg ,aes(x = reg$x, y = reg$y, color="red"),size=0.5)+theme(legend.po
sition = "none")+geom_point(data=omega ,aes(x = omega$x, y = omega$y, color="green"),size=0.5)+xlab
("x")+ylab("y")
U
```



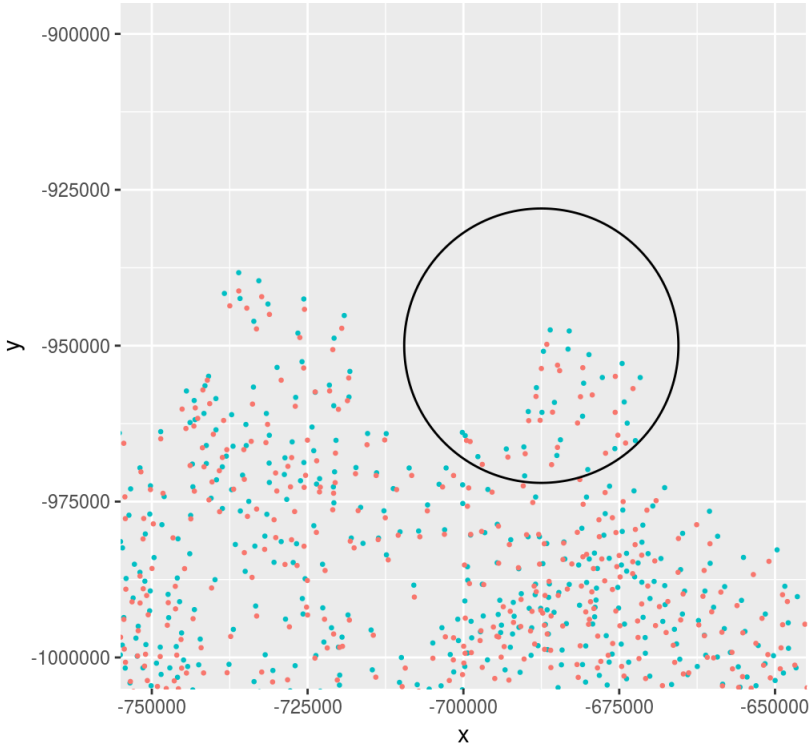
HIDE

```
U+ coord_fixed(xlim = c(-8e+5,-7e+5), ylim = c(-11e+5,-10e+5))+geom_circle(aes(x0 = -7.4e+5, y0 =
-10.45e+5, r = 22000),col="black")
```



HIDE

```
U+coord_fixed(xlim = c(-7.5e+5,-6.5e+5), ylim = c(-10e+5,-9.0e+5))+geom_circle(aes(x0 = -6.875e+5,
y0 = -9.5e+5, r = 22000),col="black")
```



Test of different package to plot Voronoi maps

ggvoronoi

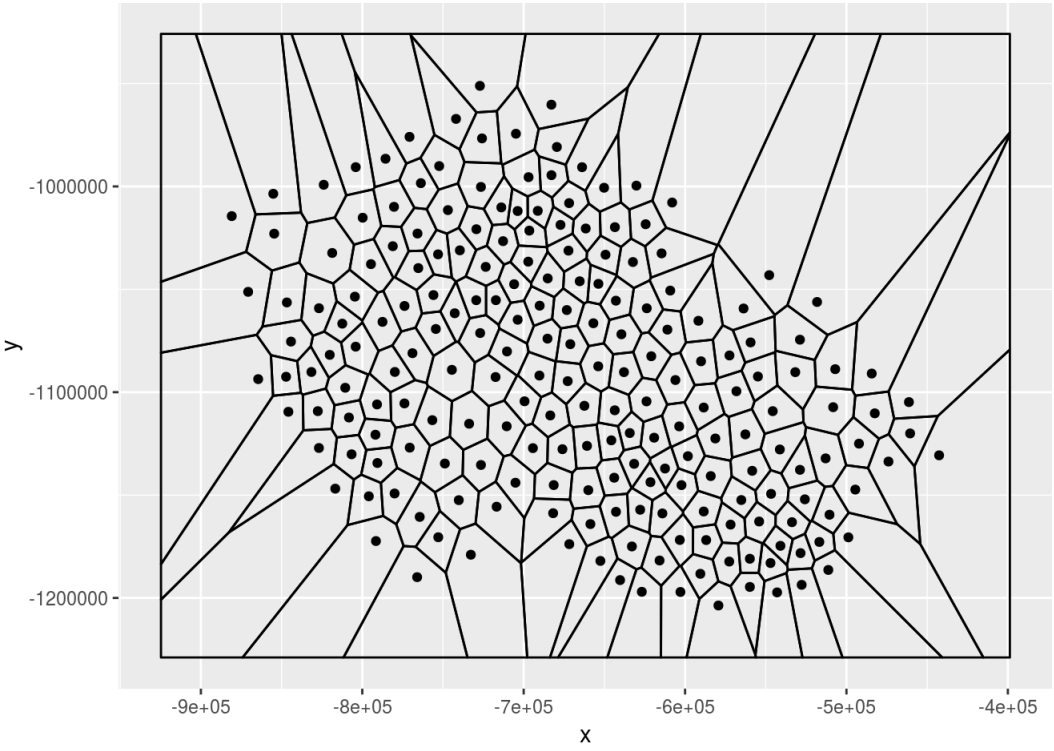
HIDE

```
print(paste0("numbers of regions :",length(unique(reg$GraOrp))))#206 regions here
```

```
## [1] "numbers of regions :206"
```

HIDE

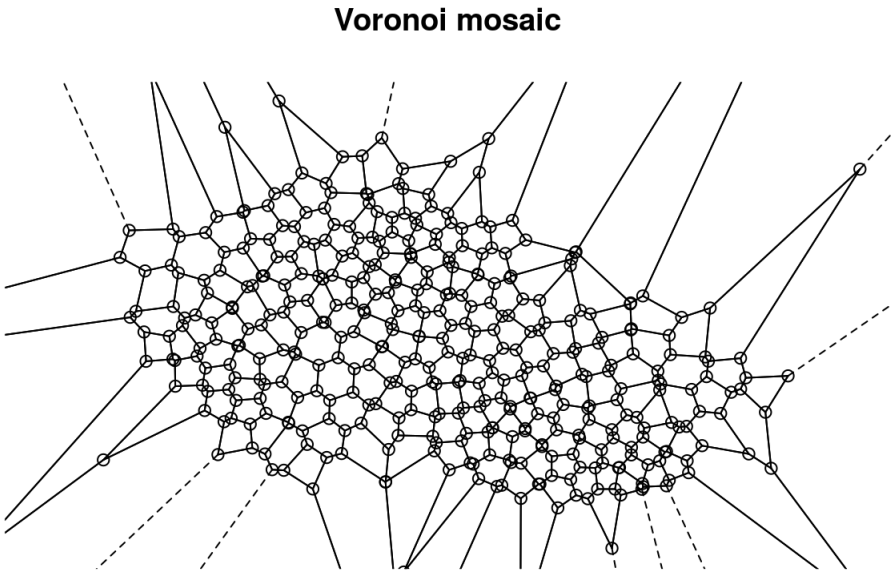
```
omega_cluster = kmeans(omega,length(unique(reg$GraOrp)))
omega_plot = ggplot(as.data.frame(omega_cluster$centers),aes(x,y))+
  stat_voronoi(geom = "path")+
  geom_point()
omega_plot
```



Tripack

HIDE

```
voronoiTripack = voronoi.mosaic(omega_cluster$centers[,1],omega_cluster$centers[,2],duplicate="error")

plot(voronoiTripack)
```



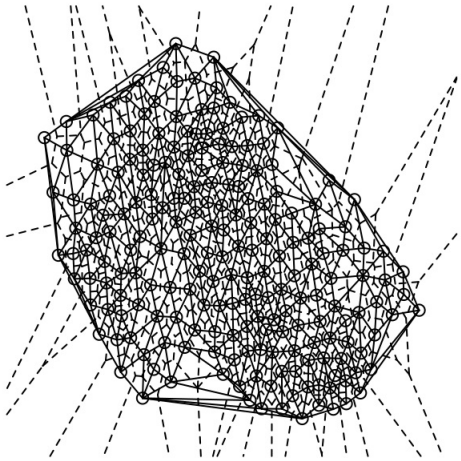
voronoiTripack

deldir

deldir

HIDE

```
x = omega_cluster$centers[,1]
y = omega_cluster$centers[,2]

voronoi_flow = deldir(x,y)
plot(voronoi_flow)
```



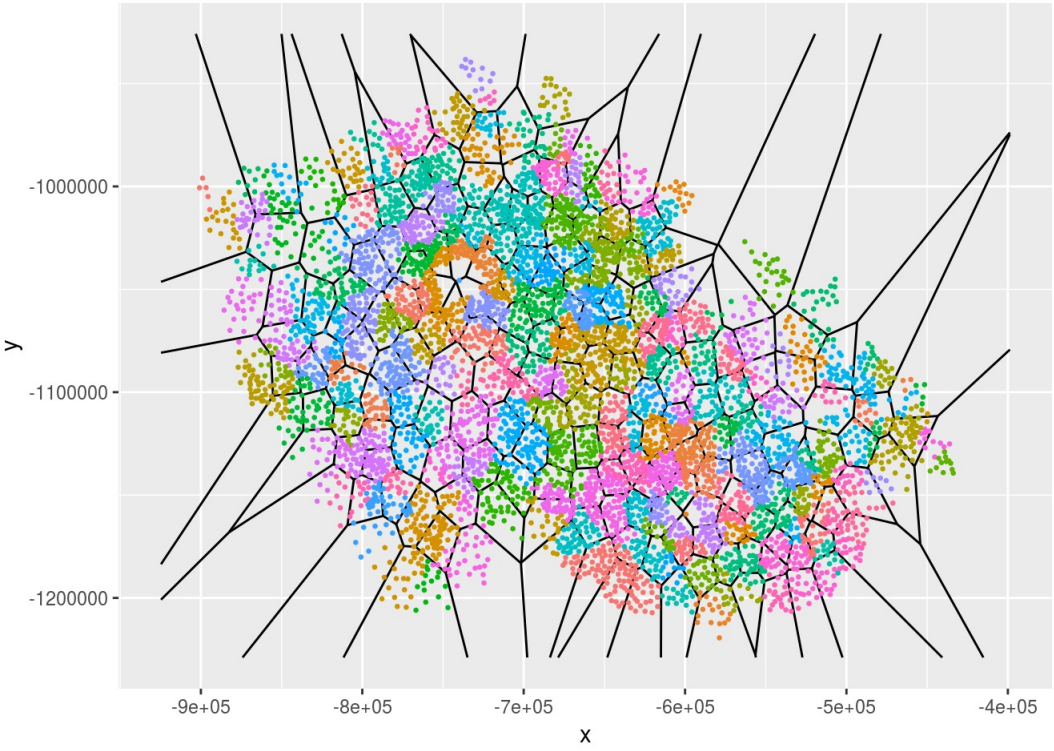
Study of different configurations of regions.

Taking flows of population and population into account.

Voronoi map of new regions and plot of cities with different color by old regions.

HIDE

```
x = omega_cluster$centers[,1]
y = omega_cluster$centers[,2]

voronoi_flow = deldir(x,y)
#The plot P is the plot of the voronoi which takes flows and populations into account
P =
  ggplot() +
  geom_segment(
    aes(x = x1, y = y1, xend = x2, yend = y2),
    size = 0.5,
    data = voronoi_flow$dirsgs,
    linetype = 1,
    alpha = 1 ) +geom_point(data=reg ,aes(x = reg$x, y = reg$y, group = reg$GraOrp, color=reg$GraOr
p),size=0.5)+theme(legend.position = "none")+xlab("x")+ylab('y')

P
```



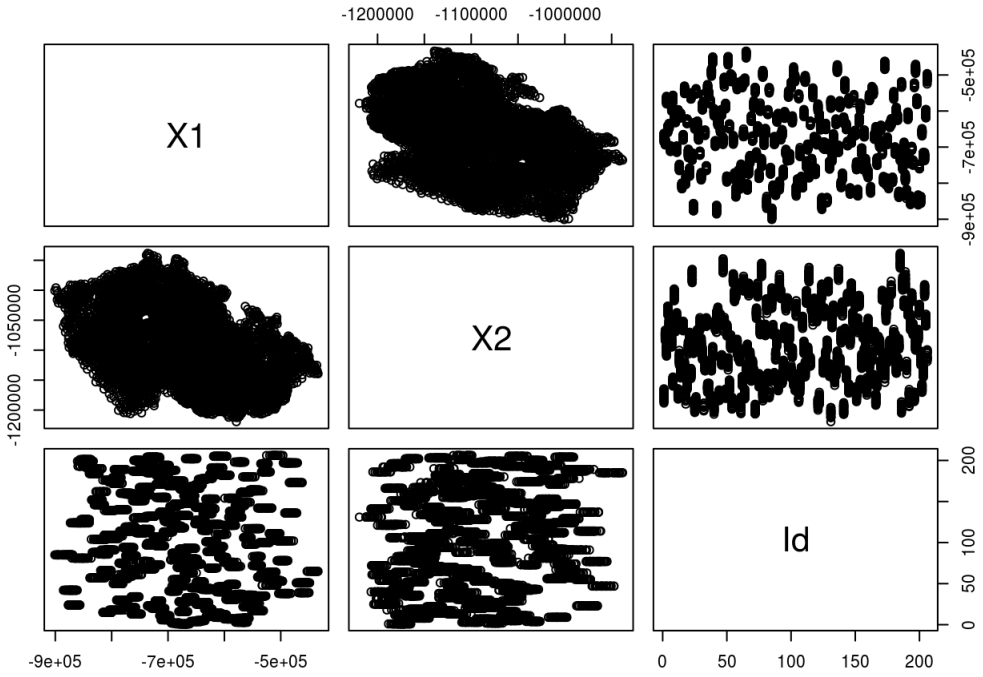
HIDE

```
#csv of the population of each voronoi region with taking the flows and populations into account
```

Creation of a CSV and statistical comparisons with old regions

HIDE

```
H= tile.list(deldir(x,y))
L=cbind(reg$x,reg$y)
V = NULL
for (i in seq(1, 206)){
  K = as.matrix(cbind(H[[i]]$x,H[[i]]$y) )
  M = pnt.in.poly(L,K)
  F = filter(M,M$pip==1)
  F$pip = NULL
  F$Id = H[[i]]$ptNum
  V = rbind(V,F)}

plot(V)
```



HIDE

```
v = na.omit(V)
v
```

1
3
4
5
6
7
8
9
10
11

1-10 of 6,257 rows | 1-1 of 4 columns

Previous **1** 2 Next

HIDE

```
f=rename(v,"x"="X1")
f=rename(f,"y"="X2")
j=reg
j$cisdan=NULL
j$ZobZkr=NULL
j$pix=NULL
j$lon=NULL
j$lat=NULL
comparison_old_regions = inner_join(j,f,by=c('x','y'))
l = NULL
for (i in seq(1,206)){
                  1
                    k = filter(comparison_old_regions,comparison_old_regions$Id==i)
                    k$region = tail(names(sort(table(k$GraOrp))), 1)
                    l=rbind(l,k)}
l$boolean = (l$GraOrp==l$region)
l=l[,c(2,3,4,1,5,6)]
l= rename(l,"new regions"="GraOrp")
l
```

x <dbl>
-665633.9
-675378.1
-673256.3
-678073.0
-677575.3
-674851.1
-672098.5
-679322.4
-686347.5
-683744.4

1-10 of 6,257 rows | 1-1 of 6 columns Previous **1** 2 Next

HIDE

```
k=(filter(l,l$boolean==TRUE))
correlation_l = length(k$x)/length(l$x)

print(paste0("city correlation = ", correlation_l))
```

```
## [1] "city correlation = 0.637525970912578"
```

HIDE

```
g=NULL
g$cities=l$`new regions`

h=NULL
h$cities=l$region
compare(g,h)
```

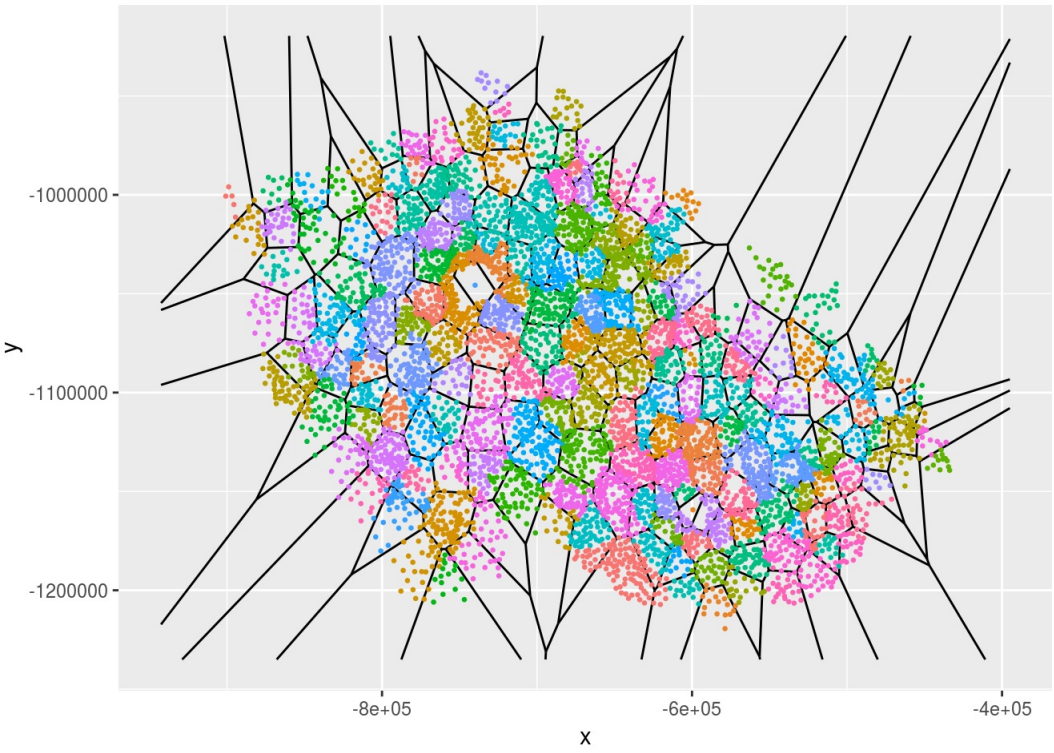
```
## Component "cities": 2268 string mismatches
```

Taking the regions that already exist (old regions)

Voronoi map highlighting the limit of this modelisation

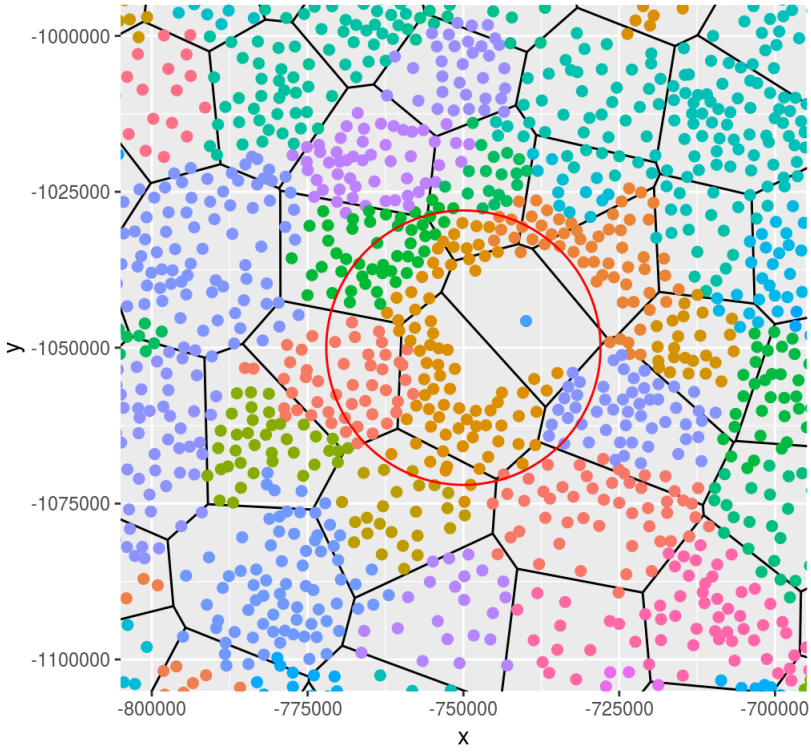
HIDE

```
voronoi_old = deldir(cisreg_l2g$x_cen,cisreg_l2g$y_cen)

Q = ggplot() +
       geom_segment(
             aes(x = x1, y = y1, xend = x2, yend = y2),
             size = 0.5,
             data = voronoi_old$dirsgs,
             linetype = 1,
             alpha = 1 ) +geom_point(data=reg ,aes(x = reg$x, y = reg$y, group = reg$GraOrp, color
=reg$GraOrp),size=0.5)+theme(legend.position = "none")+xlab("x")+ylab('y')
Q
```



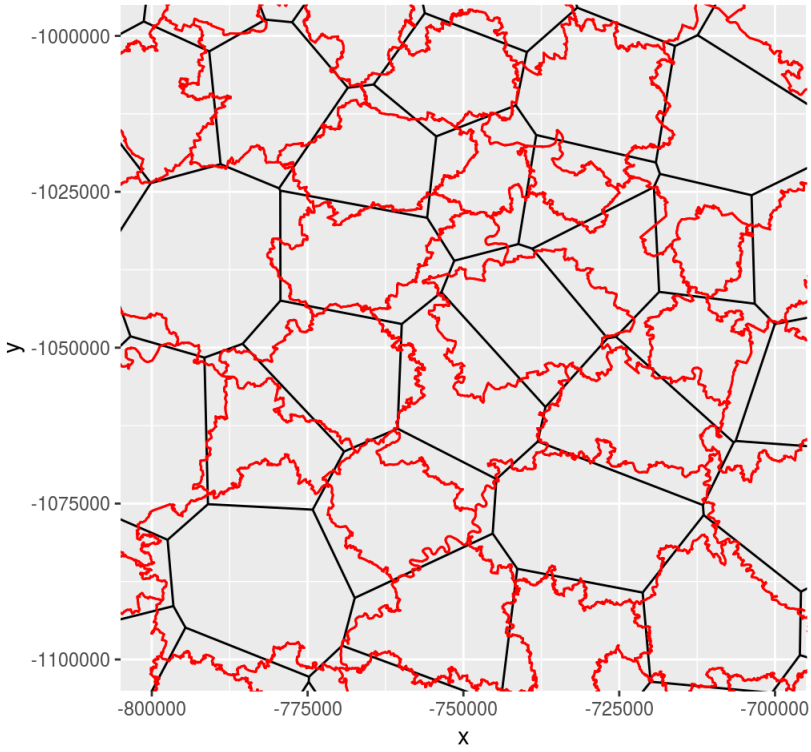
HIDE

```
Q = ggplot() +
       geom_segment(
             aes(x = x1, y = y1, xend = x2, yend = y2),
             size = 0.5,
             data = voronoi_old$dirsgs,
             linetype = 1,
             alpha = 1 ) +geom_point(data=reg ,aes(x = reg$x, y = reg$y, group = reg$GraOrp, color
=reg$GraOrp),size=2)+theme(legend.position = "none") +xlab("x")+ylab('y')

Q <- Q + coord_fixed(xlim = c(-8e+5,-7e+5), ylim = c(-11e+5,-10e+5))+geom_circle(aes(x0 = -7.5e+5,
y0 = -10.5e+5, r = 22000),col="red")

Q
```



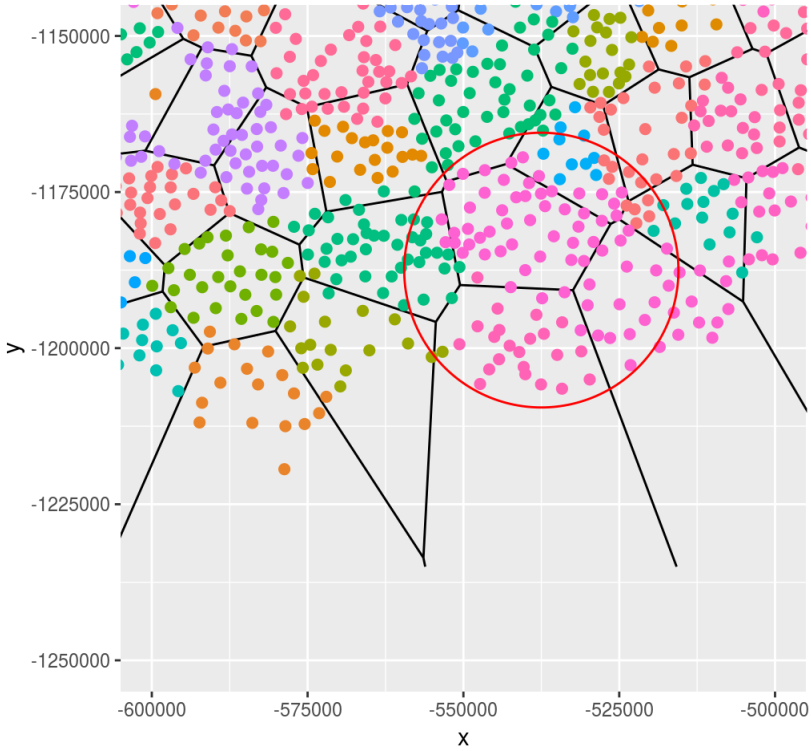
HIDE

```
Q = ggplot() +
       geom_segment(
             aes(x = x1, y = y1, xend = x2, yend = y2),
             size = 0.5,
             data = voronoi_old$dirsgs,
             linetype = 1,
             alpha = 1 )+ geom_polygon(data =intpol_pro_s3e_vsb_sacz0_com_geomapfor_l2g_pol, aes(x
= x, y = y, group = group), color = "red", size = 0.5, fill = NA)+ coord_fixed(xlim = c(-8e+5,-7e+5
), ylim = c(-11e+5,-10e+5))+xlab("x")+ylab('y')
Q
```



HIDE

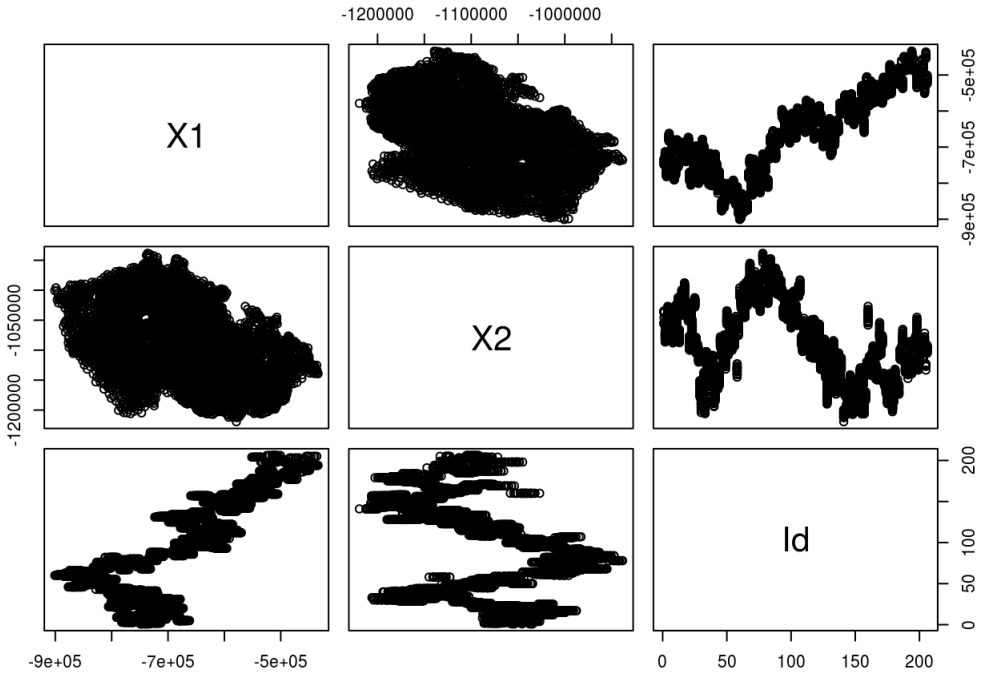
```
Q = ggplot() +
       geom_segment(
             aes(x = x1, y = y1, xend = x2, yend = y2),
             size = 0.5,
             data = voronoi_old$dirsgs,
             linetype = 1,
             alpha = 1 ) +geom_point(data=reg ,aes(x = reg$x, y = reg$y, group = reg$GraOrp, color
=reg$GraOrp),size=2)+theme(legend.position = "none") +xlab("x")+ylab('y')

Q <- Q +geom_circle(aes(x0 = -537500, y0 = -1187500, r = 22000),col="red")+ coord_fixed(xlim = c(-
6e+5,-5e+5), ylim = c(-12.5e+5,-11.5e+5))
Q
```



Creation of a CSV and statistical comparisons with old regions

HIDE

```
HH= tile.list(voronoi_old)
LL=cbind(reg$x,reg$y)
VV = NULL
for (i in seq(1, 206)){
  KK = as.matrix(cbind(HH[[i]]$x,HH[[i]]$y) )
  MM = pnt.in.poly(LL,KK)
  FF = filter(MM,MM$pip==1)
  FF$pip = NULL
  FF$Id = HH[[i]]$ptNum
  VV = rbind(VV,FF)}

plot(VV)
```



HIDE

```
vv = na.omit(VV)
vv
```

1
3
4
5
6
7
8
9
10
11

1-10 of 6,257 rows | 1-1 of 4 columns

Previous **1** 2 Next

HIDE

```
f=rename(vv,"x"="X1")
f=rename(f,"y"="X2")
j=reg
j$cisdan=NULL
j$ZobZkr=NULL
j$pix=NULL
j$lon=NULL
j$lat=NULL
comparison_old_regions = inner_join(j,f,by=c('x','y'))
ll = NULL
for (i in seq(1,206)){
                  1
                    kk = filter(comparison_old_regions,comparison_old_regions$Id==i)
                    kk$region = tail(names(sort(table(kk$GraOrp))), 1)
                    ll=rbind(ll,kk)}
ll$boolean = (ll$GraOrp==ll$region)
ll=ll[,c(2,3,4,1,5,6)]
ll= rename(ll,"new regions"="GraOrp")
ll
```

x <dbl>
-738645.9
-732454.7
-740811.7
-748081.1
-736647.4
-734919.1
-744489.8
-749076.1
-746545.3
-739951.5

1-10 of 6,257 rows | 1-1 of 6 columns Previous 1 2 Next

HIDE

```
kk=(filter(ll,ll$boolean==TRUE))
correlation_ll = length(kk$x)/length(ll$x)

print(paste0("city correlation = ", correlation_ll))
```

```
## [1] "city correlation = 0.77688988333067"
```

HIDE

```
g=NULL
g$cities=ll$`new regions`

h=NULL
h$cities=ll$region
compare(g,h)
```

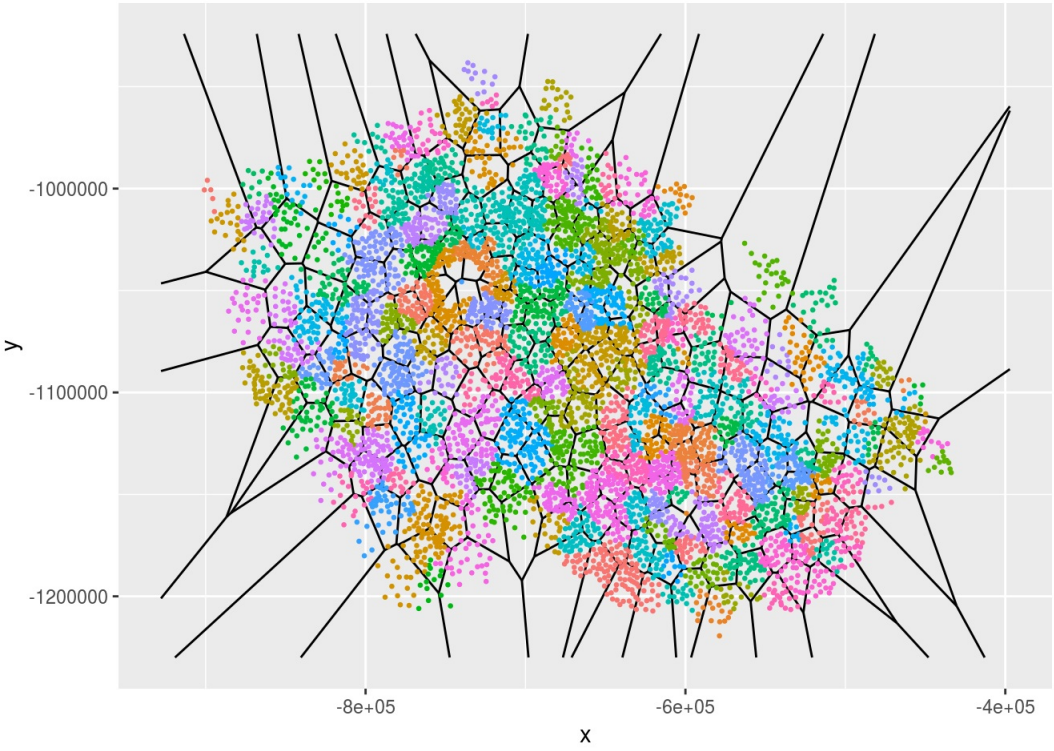
```
## Component "cities": 1396 string mismatches
```

Most fair configuration possible

Voronoi map

HIDE

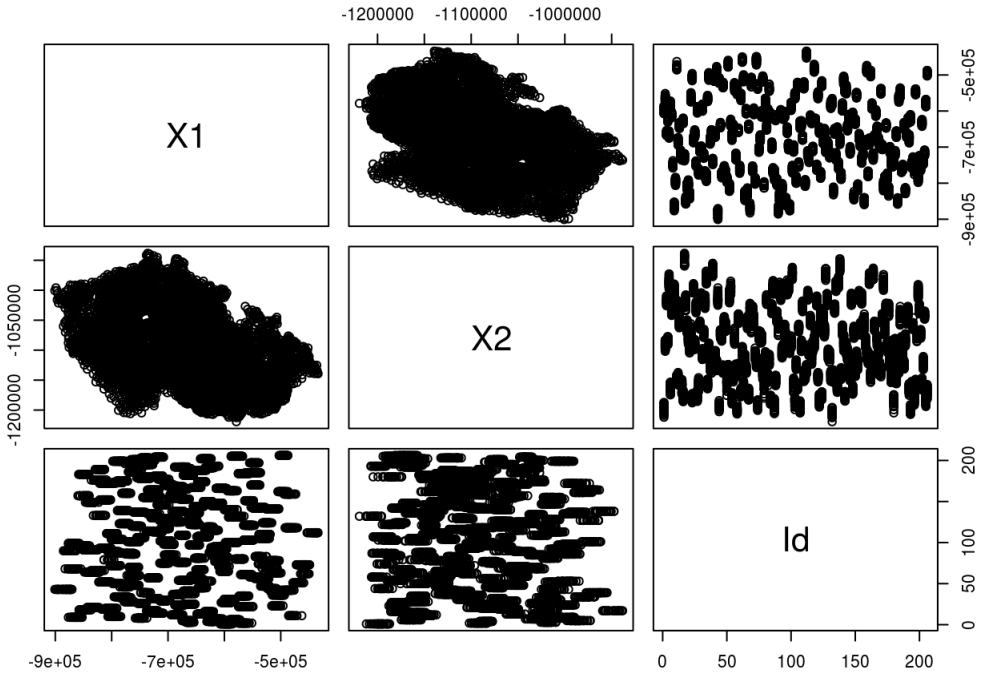
```
fair = kmeans(na.omit(as.data.frame(cbind(reg$x,reg$y))),length(unique(reg$GraOrp)))
X = fair$centers[,1]
Y = fair$centers[,2]
voronoi_fair = deldir(X,Y)

#The plot R is the plot of the voronoi in  the most fair configuration
R =
  ggplot() +
  geom_segment(
    aes(x = x1, y = y1, xend = x2, yend = y2),
    size = 0.5,
    data = voronoi_fair$dirsgs,
    linetype = 1,
    alpha = 1 ) +geom_point(data=reg ,aes(x = reg$x, y = reg$y, group = reg$GraOrp, color=reg$GraOr
p),size=0.5)+theme(legend.position = "none")+xlab("x")+ylab('y')
R
```



Creation of a CSV and statistical comparisons with old regions

HIDE

```
HHH = tile.list(deldir(X,Y))
LLL=cbind(reg$x,reg$y)
VVV = NULL
for (i in seq(1, 206)){
  KKK = as.matrix(cbind(HHH[[i]]$x,HHH[[i]]$y) )
  MMM = pnt.in.poly(LLL,KKK)
  FFF = filter(MMM,MMM$pip==1)
  FFF$pip = NULL
  FFF$Id = HHH[[i]]$ptNum
  VVV = rbind(VVV,FFF)}

plot(VVV)
```



HIDE

```
vvv = na.omit(VVV)
vvv
```

2
3
4
5
6
7
8
9
10
11

1-10 of 6,257 rows | 1-1 of 4 columns

HIDE

```
#csv of the population of each voronoi region created by the most fair configuration

f=rename(vvv,"x"="X1")
f=rename(f,"y"="X2")
j=reg
j$cisdan=NULL
j$ZobZkr=NULL
j$pix=NULL
j$lon=NULL
j$lat=NULL
comparison_old_regions = inner_join(j,f,by=c('x','y'))
lll = NULL
for (i in seq(1,206)){
                    kkk = filter(comparison_old_regions,comparison_old_regions$Id==i)
                    kkk$region = tail(names(sort(table(kkk$GraOrp))), 1)
                    lll=rbind(lll,kkk)}
lll$boolean = (lll$GraOrp==lll$region)
lll=lll[,c(2,3,4,1,5,6)]
lll= rename(lll,"new regions"="GraOrp")
lll
```

x <dbl>
-589341.5
-601979.1
-607207.0
-605419.6
-593109.1
-607141.6
-604591.0
-599360.5
-591934.9
-601431.1

1-10 of 6,257 rows | 1-1 of 6 columns Previous 1 2 Next

HIDE

```
kkk=(filter(lll,lll$boolean==TRUE))
correlation_lll = length(kkk$x)/length(lll$x)
print(paste0("city correlation = ", correlation_lll))
```

```
## [1] "city correlation = 0.620744765862234"
```

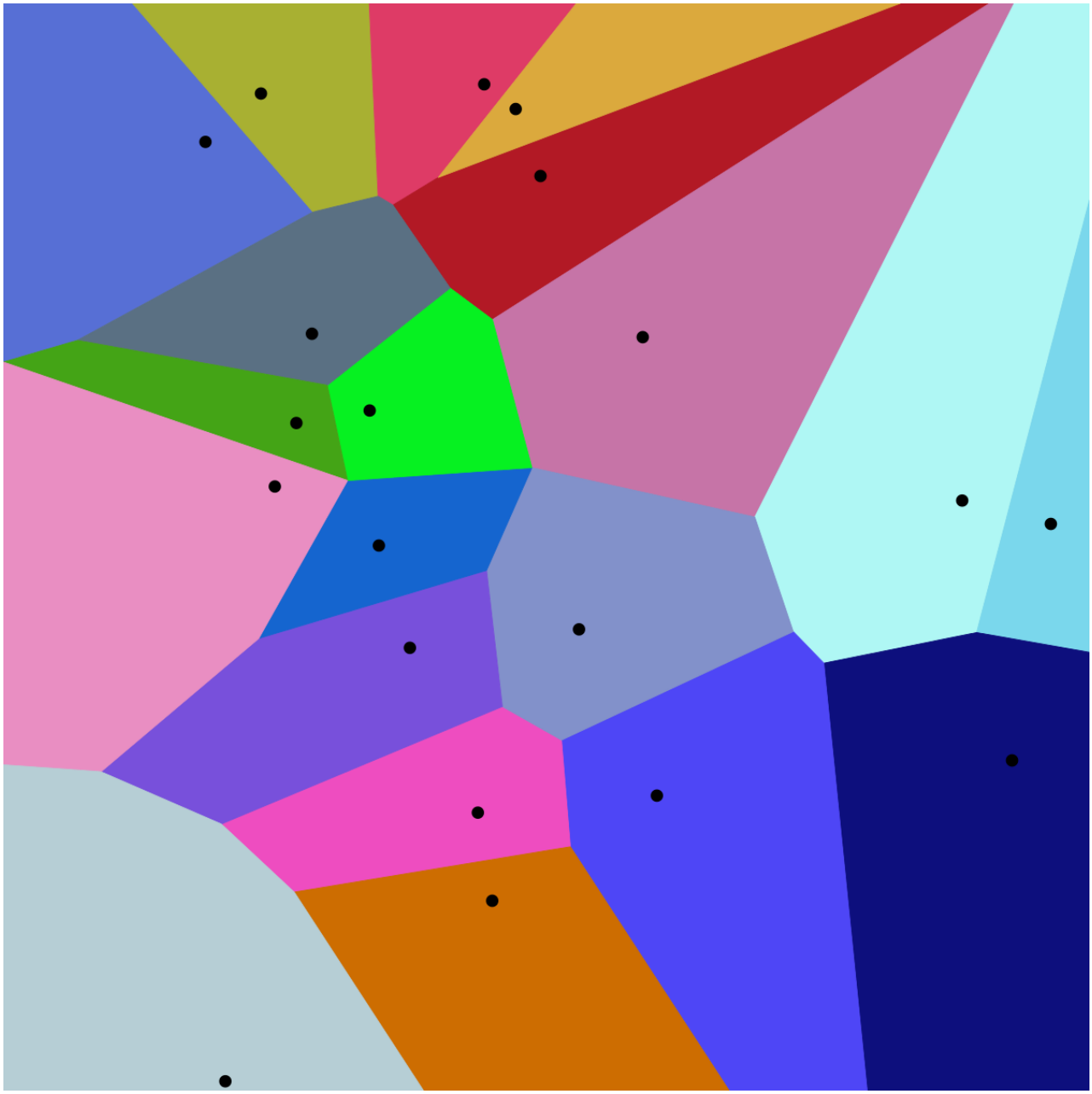
HIDE

```
g=NULL
g$cities=lll$`new regions`

h=NULL
h$cities=lll$region
compare(g,h)
```

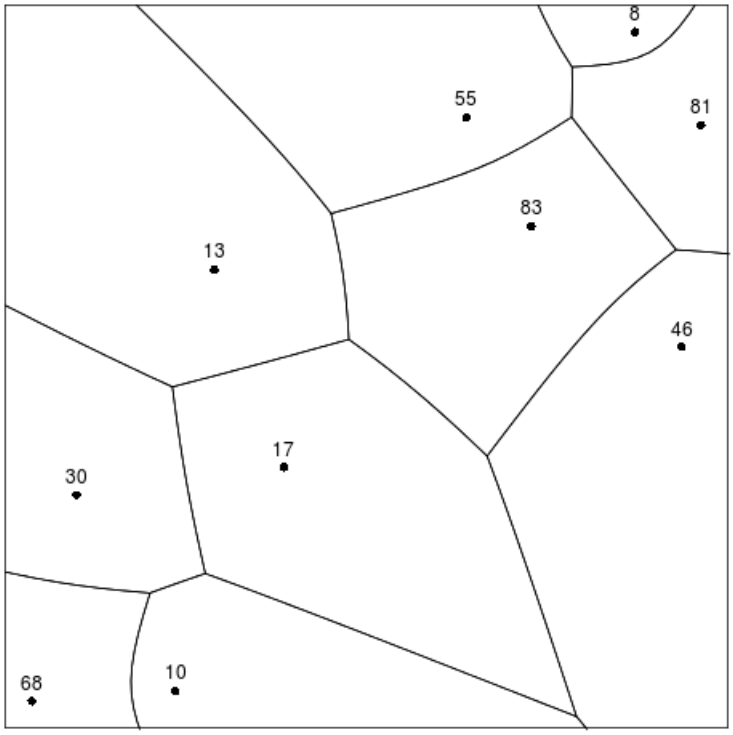
```
## Component "cities": 2373 string mismatches
```

Attempt of improvement

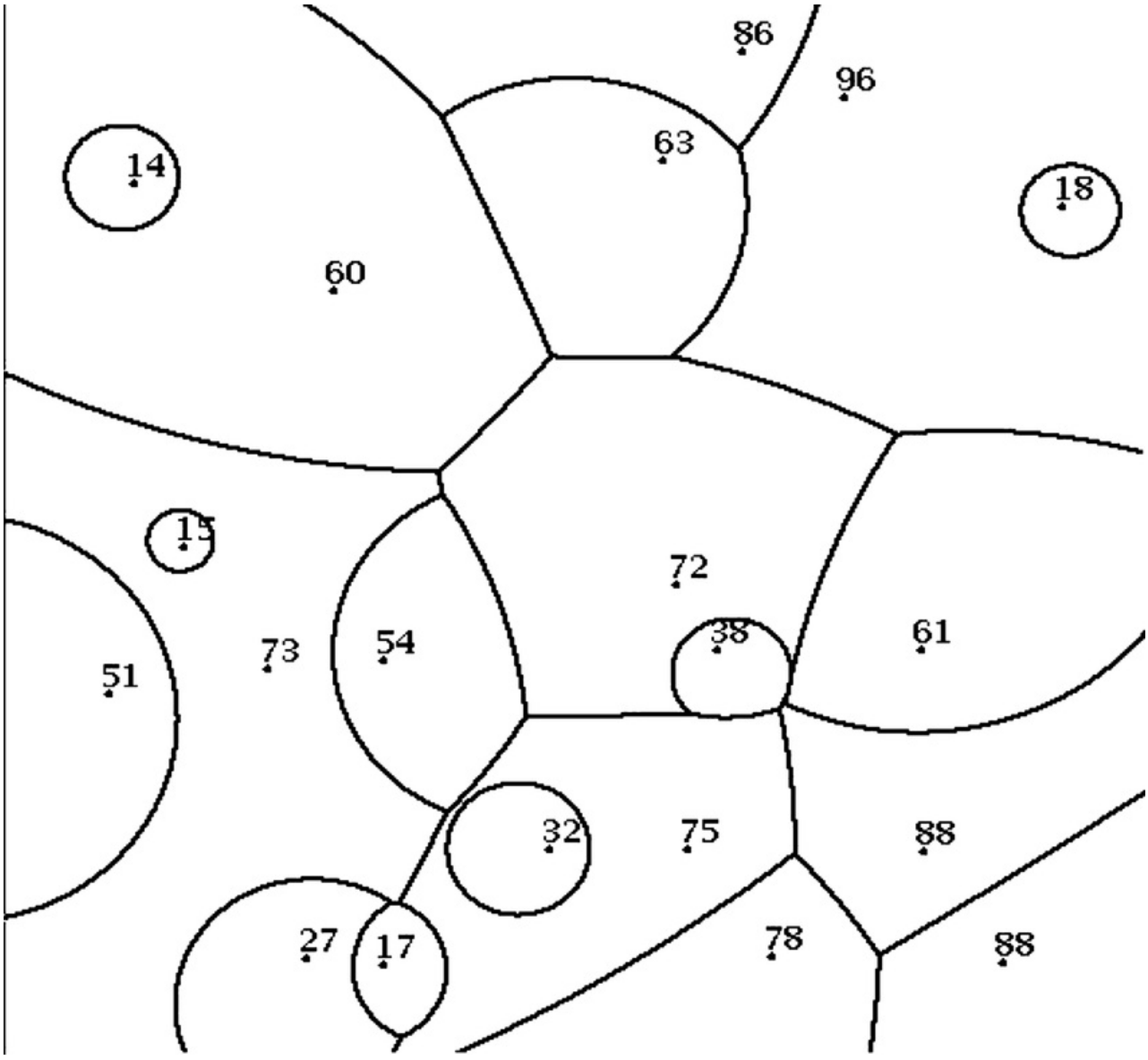
Weighted Voronoi



“Standard Voronoi”



“additively weighted Voronoi”



“multiplicatively weighted Voronoi”

Taking limitation of distance into account

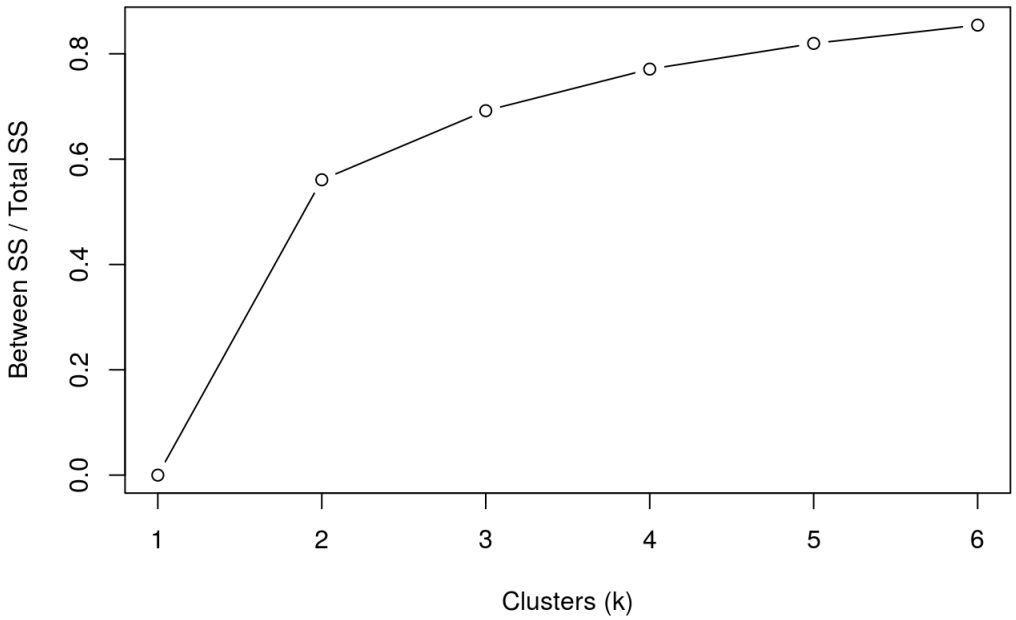
Clu002

Without further parameter

X-means clustering

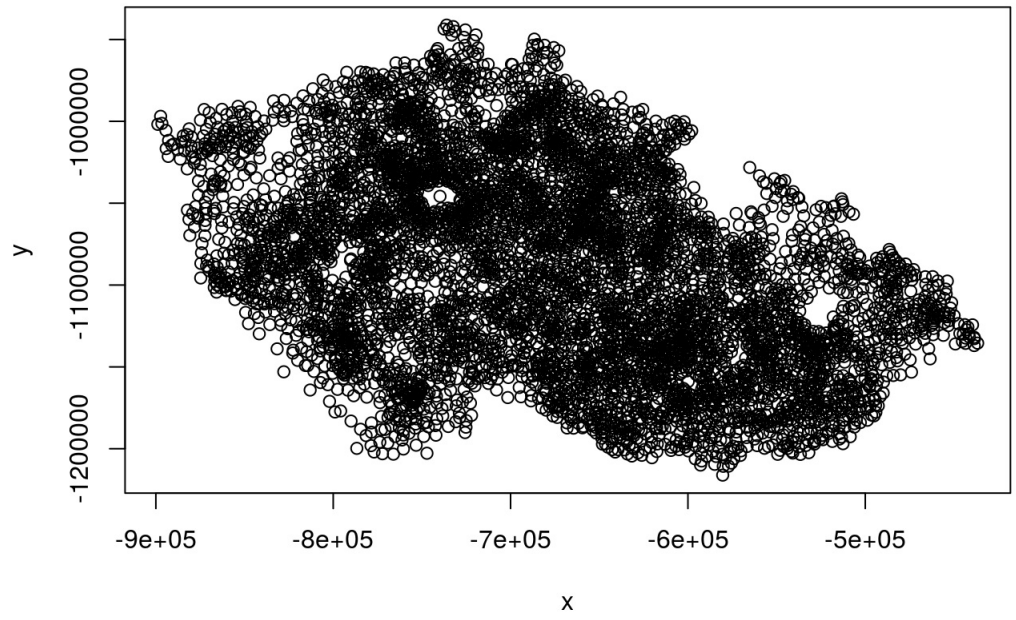
HIDE

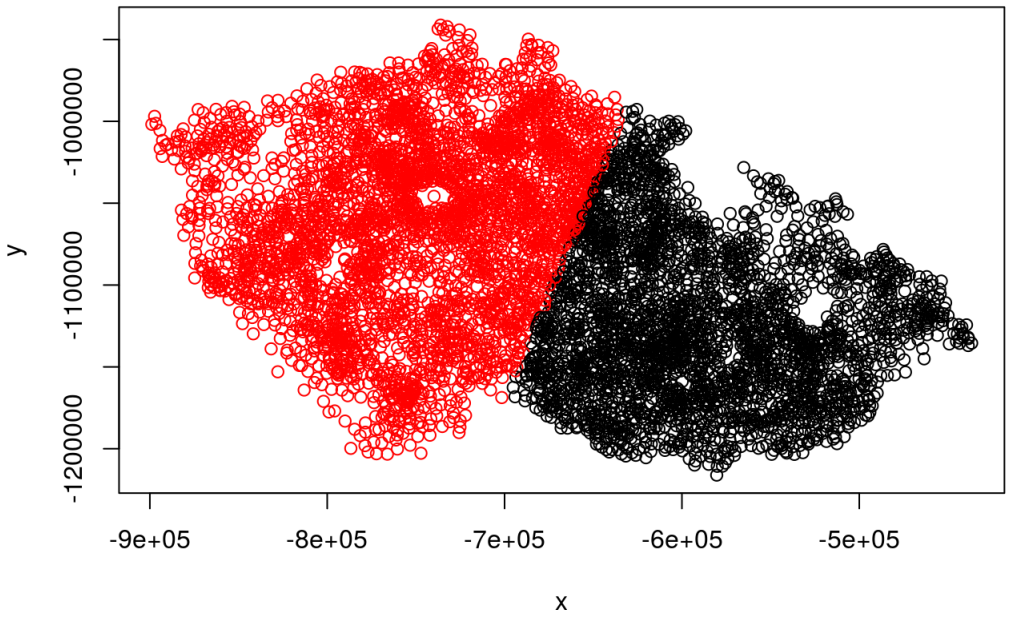
```
## CHOOSING K
k <- list()
for(i in 1:6){
  k[[i]] <- kmeans(omega, i)
}

#k

betweenss_totss <- list()
for(i in 1:6){
  betweenss_totss[[i]] <- k[[i]]$betweenss/k[[i]]$totss
}

plot(1:6, betweenss_totss, type = "b",
     ylab = "Between SS / Total SS", xlab = "Clusters (k)")
```

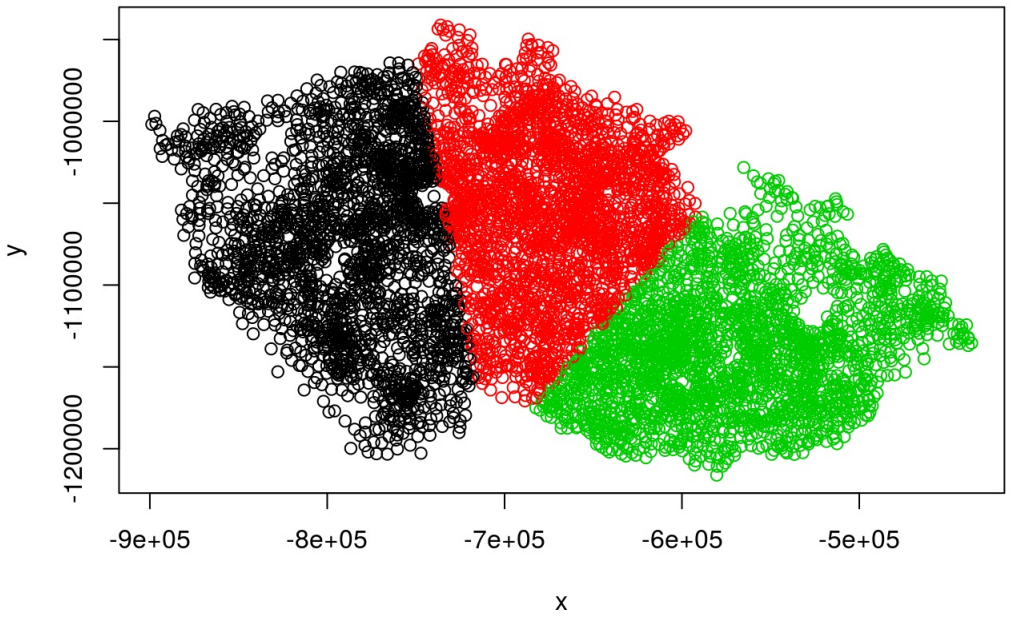


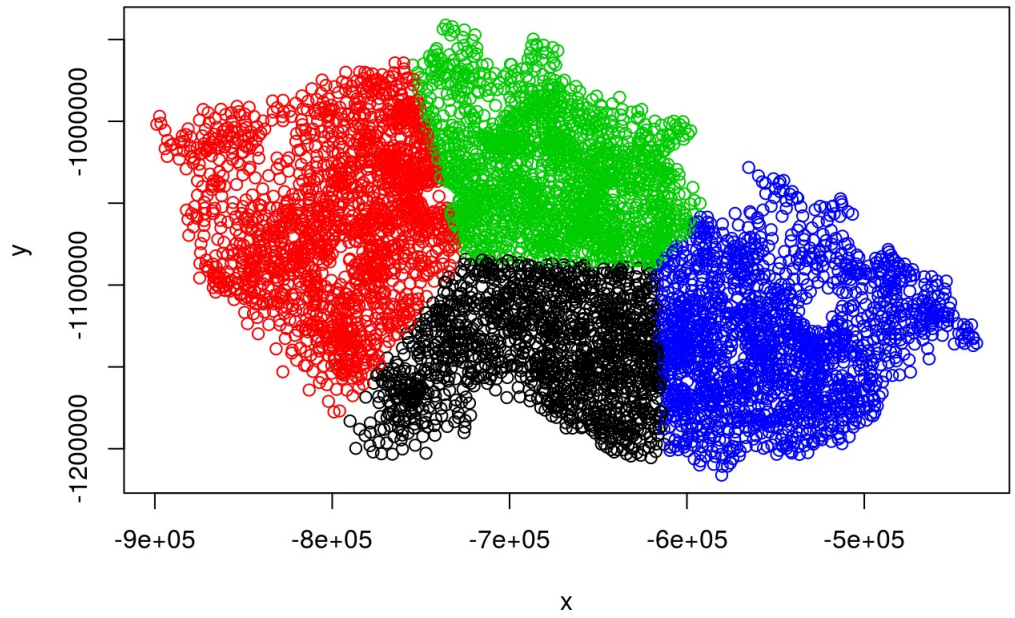
HIDE

```
for(i in 1:4){
  plot(omega, col = k[[i]]$cluster)
}
```




y
-1000000
-1100000
-1200000
-9e+05
-8e+05
-7e+05
-6e+05
-5e+05
x


y
-1000000
-1100000
-1200000
-9e+05
-8e+05
-7e+05
-6e+05
-5e+05
x

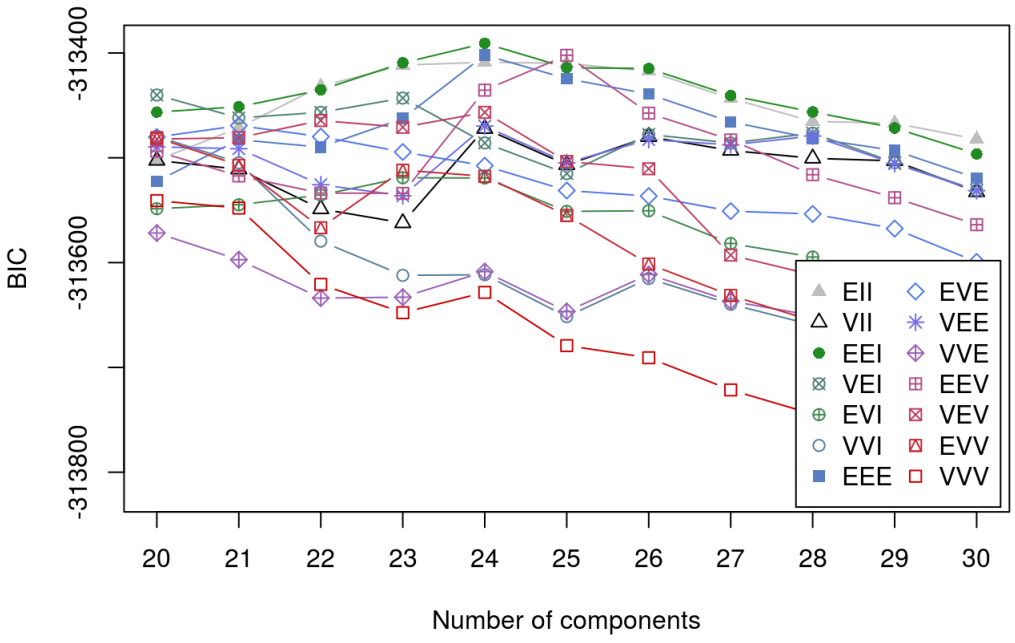


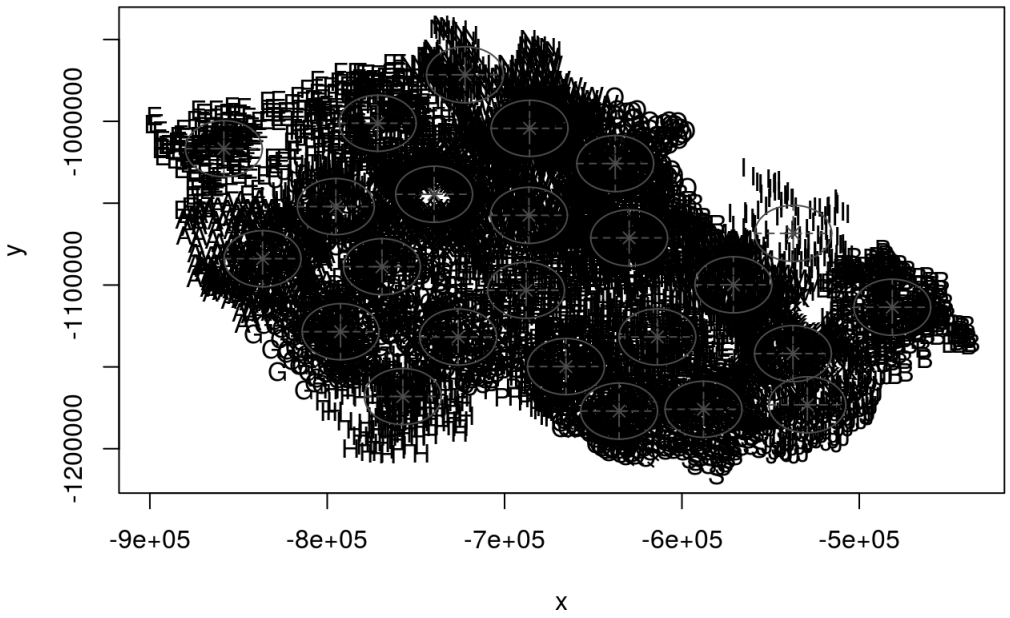
EM algorithm

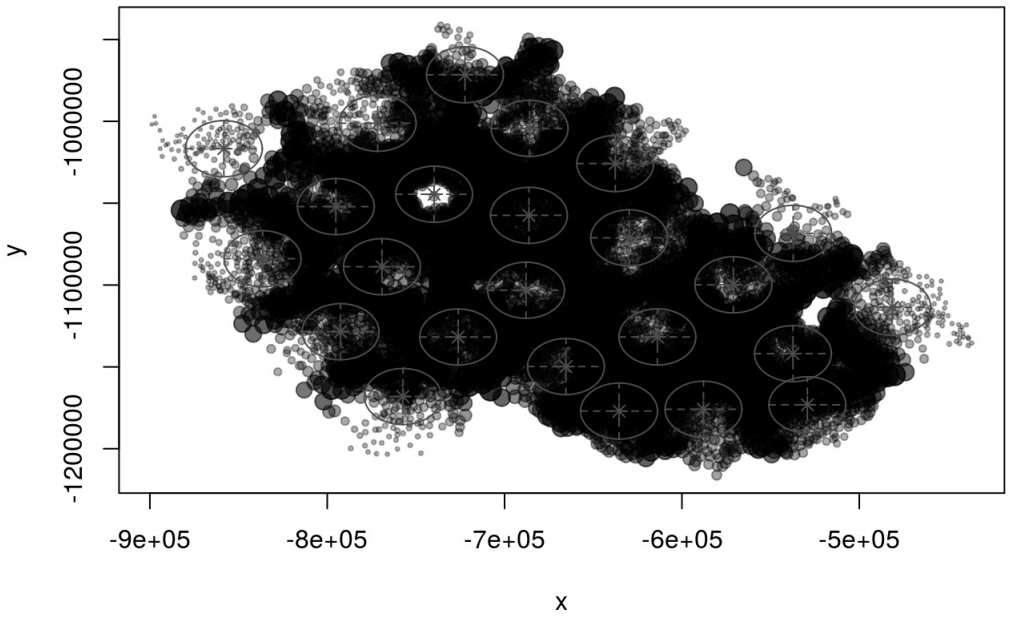
BIC criterion

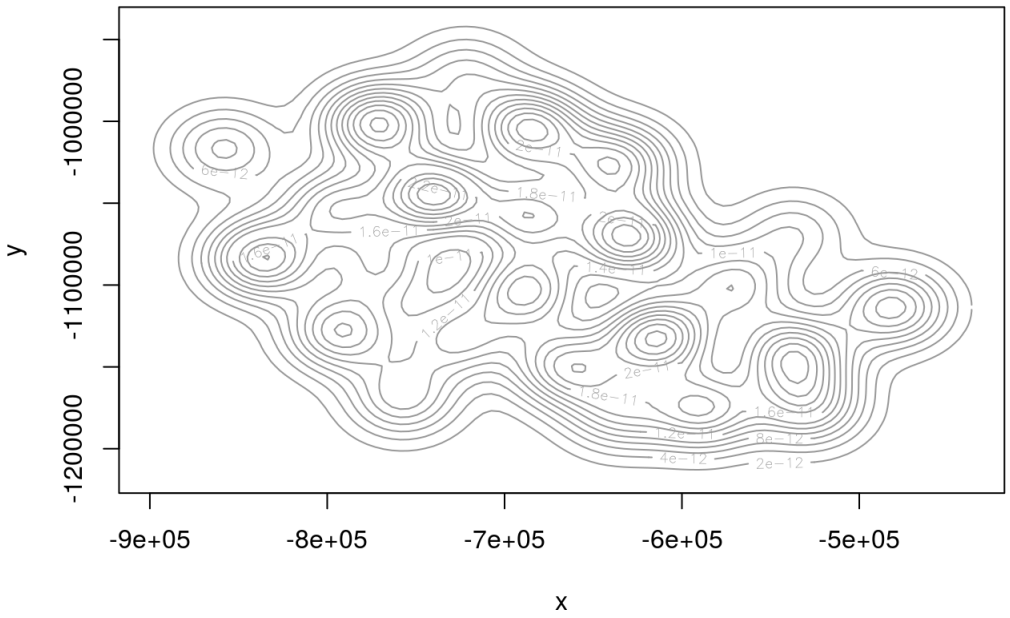
HIDE

```
# MODEL-BASED CLUSTERING ----
fitM <- Mclust(omega,G=20:30)
plot(fitM)
```




y
-1000000
-1100000
-1200000
-9e+05
-8e+05
-7e+05
-6e+05
-5e+05
x

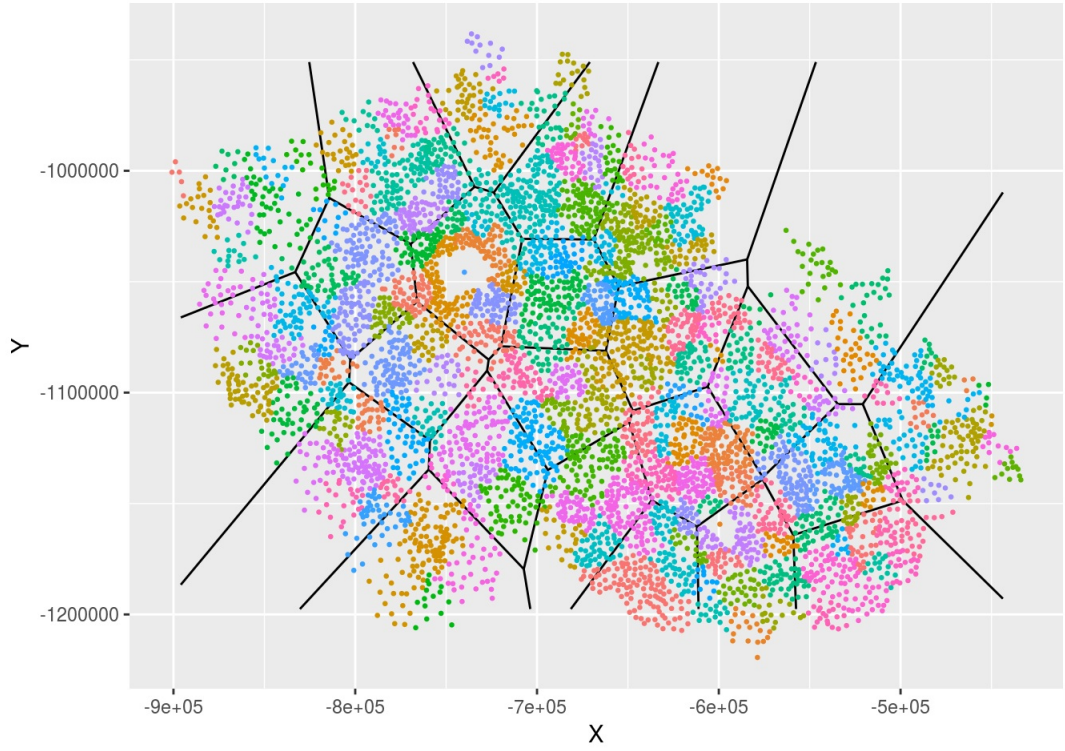

y
-1000000
-1100000
-1200000
-9e+05
-8e+05
-7e+05
-6e+05
-5e+05
x



Voronoi map

HIDE

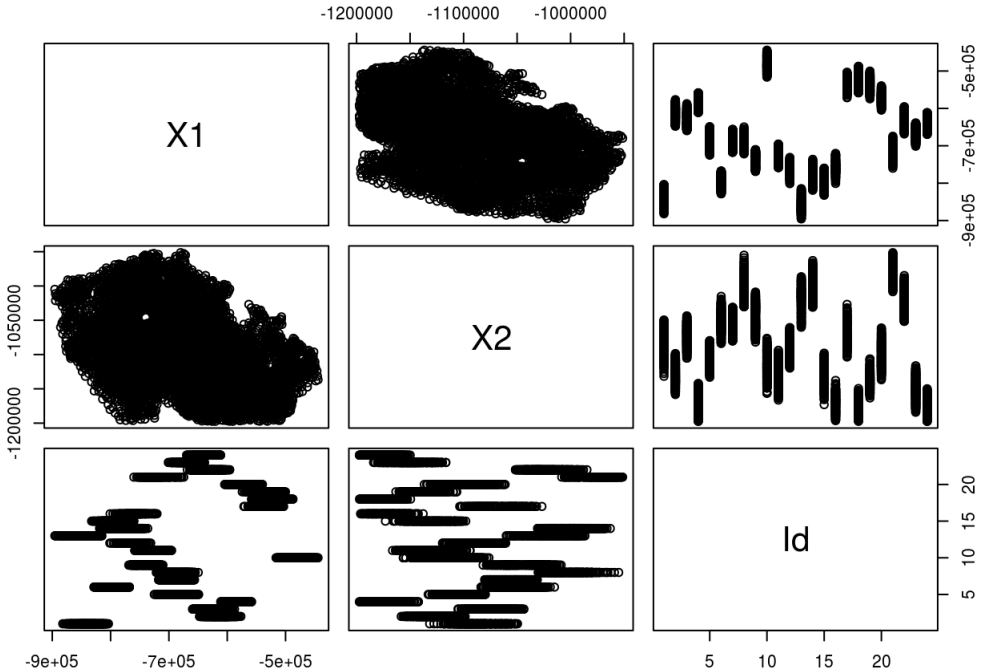
```
F = t(as.matrix(fitM$parameters$mean))
F = deldir(as.data.frame(F))
#plotting method ----
p.ele <- ggplot() #+ theme(aspect.ratio = (asp.rat.ext))
#p.ele <- p.ele + coord_fixed(xlim = c(-8e+5,-7e+5), ylim = c(-12e+5,-10e+5))

# FOR POLYGONS
# FOR POINTS
#p.ele <- p.ele + geom_point(data = attraseva.adm.map.all, aes(lon,lat), size=0.5, colour="seashell
1",shape=15,alpha=0.4)

p.ele = p.ele+ geom_segment(
  aes(x = x1, y = y1, xend = x2, yend = y2),
  size = 0.5,
  data = F$dirsgs,
  linetype = 1,
  alpha = 1 ) +geom_point(data=reg ,aes(x = reg$x, y = reg$y, group = reg$GraOrp, color=reg$GraOrp)
,size=0.5)+theme(legend.position = "none")
#p.ele <- p.ele + theme.clean.F
p.ele <- p.ele + labs(
  #title="Voronoi Map of regions",
  #subtitle="",
  x = "X",y = "Y")

#p.ele <- p.ele + geom_polygon(data =intpol_pro_s3e_vsb_sacz0_com_geomapfor_12g_pol, aes(x = x, y =
y, group = group), color = "black", size = 0.5, fill = NA)
p.ele
```



Creation of a CSV and statistical comparisons with old regions

HIDE

```
HHH = tile.list(F)
LLL=cbind(reg$x,reg$y)
VVV = NULL
for (i in seq(1, F$n.data)){
  KKK = as.matrix(cbind(HHH[[i]]$x,HHH[[i]]$y) )
  MMM = pnt.in.poly(LLL,KKK)
  FFF = filter(MMM,MMM$pip==1)
  FFF$pip = NULL
  FFF$Id = HHH[[i]]$ptNum
  VVV = rbind(VVV,FFF)}

plot(VVV)
```



HIDE

```
vvv = na.omit(VVV)
vvv
```

1
2
3
4
5
6
8
9
10
11

1-10 of 6,135 rows | 1-1 of 4 columns Previous **1** 2 Next

HIDE

```
#csv of the population of each voronoi region created by the most fair configuration

f=rename(vvv,"x"="X1")
f=rename(f,"y"="X2")
j=reg
j$cisdan=NULL
j$ZobZkr=NULL
j$pix=NULL
j$lon=NULL
j$lat=NULL
comparison_old_regions = inner_join(j,f,by=c('x','y'))
lll = NULL
for (i in seq(1,F$n.data)){
                    kkk = filter(comparison_old_regions,comparison_old_regions$Id==i)
                    kkk$region = tail(names(sort(table(kkk$GraOrp))), 1)
                    lll=rbind(lll,kkk)}
lll$boolean = (lll$GraOrp==lll$region)
lll=lll[,c(2,3,4,1,5,6)]
lll= rename(lll,"new regions"="GraOrp")
lll
```

x <dbl>
-848610.6
-836178.1
-810002.1
-812844.7
-842397.6
-840342.2
-840088.8
-813493.8
-820024.7
-832562.7

1-10 of 6,135 rows | 1-1 of 6 columns Previous **1** 2 Next

HIDE

```
kkk=(filter(lll,lll$boolean==TRUE))
correlation_lll = length(kkk$x)/length(lll$x)
print(paste0("city correlation = ", correlation_lll))
```

```
## [1] "city correlation = 0.234229828850856"
```

HIDE

```
g=NULL
g$cities=lll$`new regions`

h=NULL
h$cities=lll$region
compare(g,h)
```

```
## Component "cities": 4698 string mismatches
```

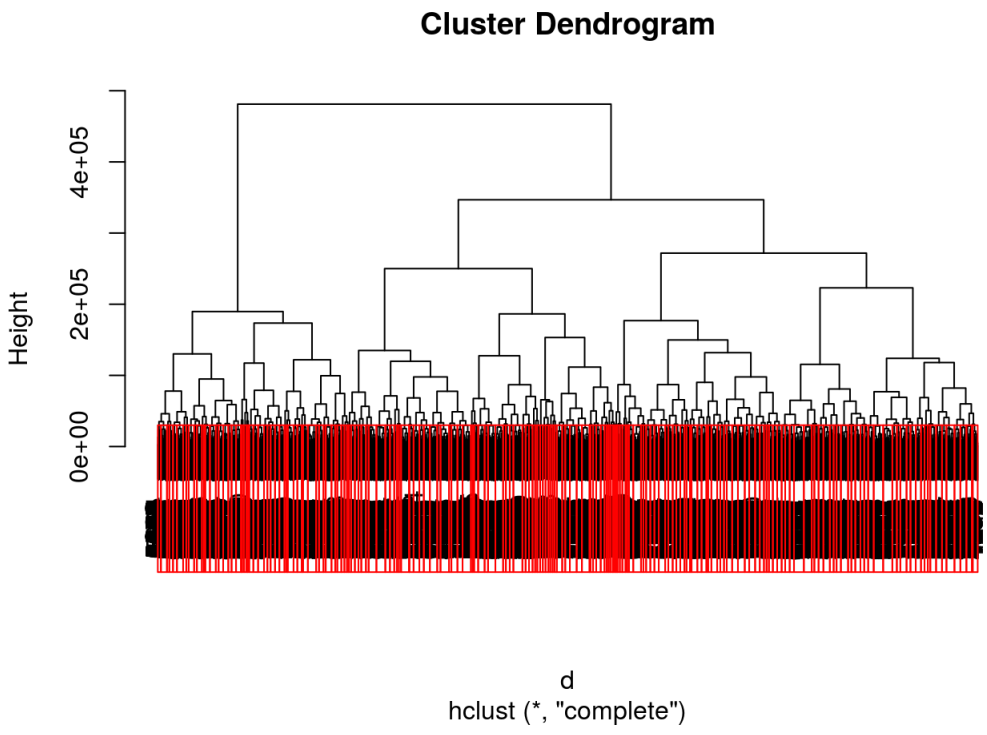
With limitation of distance between services centers and cities

Hierarchical clustering

Dendrogram and clustered plot

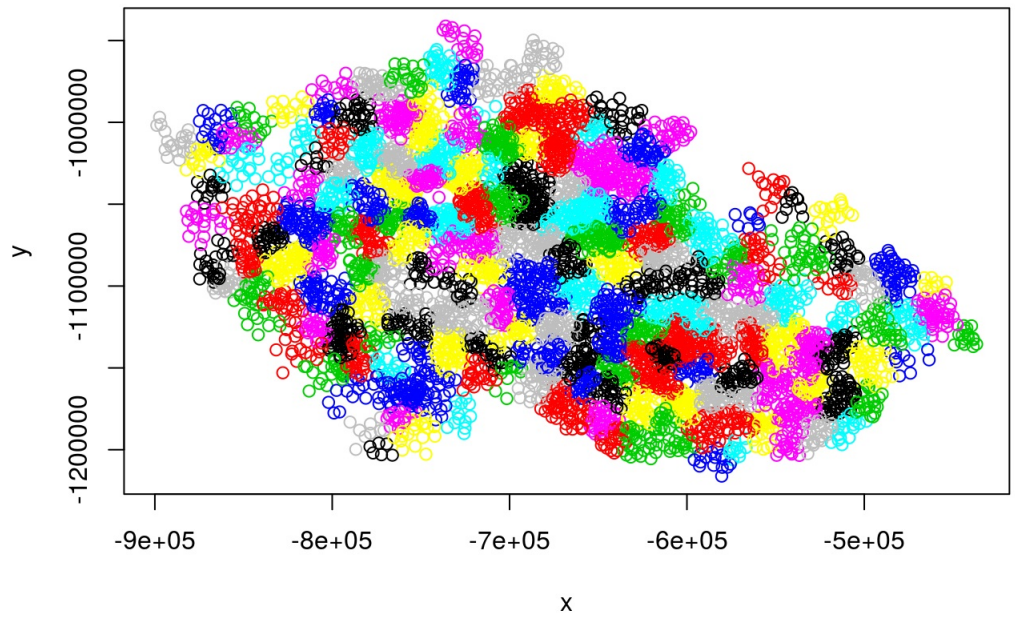
HIDE

```
# HIERACHICAL CLUSTERING ----
distance.max = 15000
d <- dist(omega)
fitH <- hclust(d, "complete")
plot(fitH)
rect.hclust(fitH, h = 2*distance.max , border = "red")
```



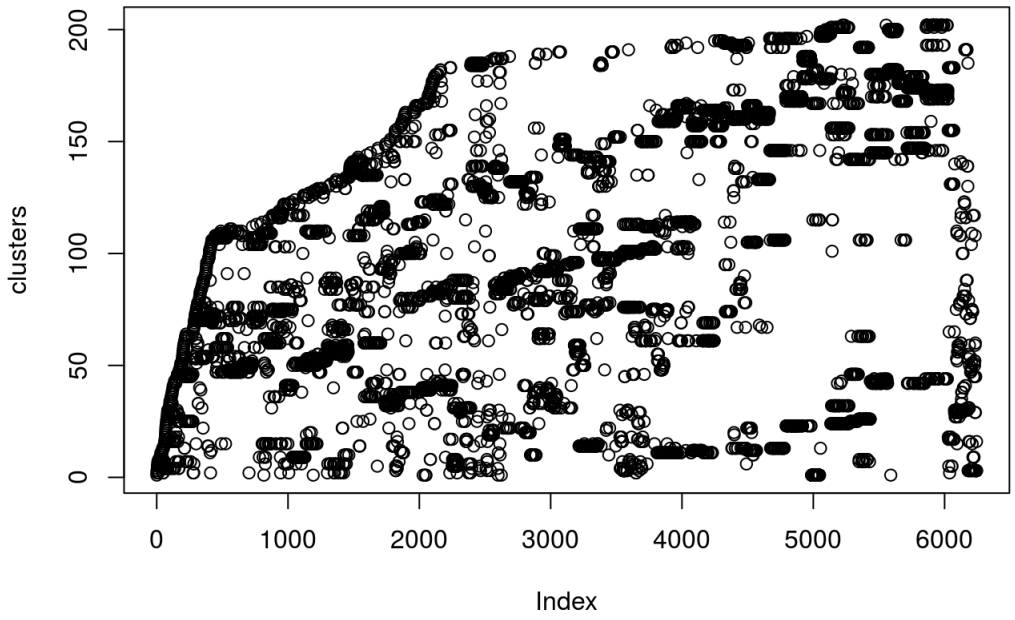
HIDE

```
clusters <- cutree(fitH, h=2*distance.max)
plot(omega, col = clusters)
```



HIDE

```
plot(clusters)
```



HIDE

```
summary(clusters)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##    1.00   47.00   93.00   96.12  149.00  202.00
```

HIDE

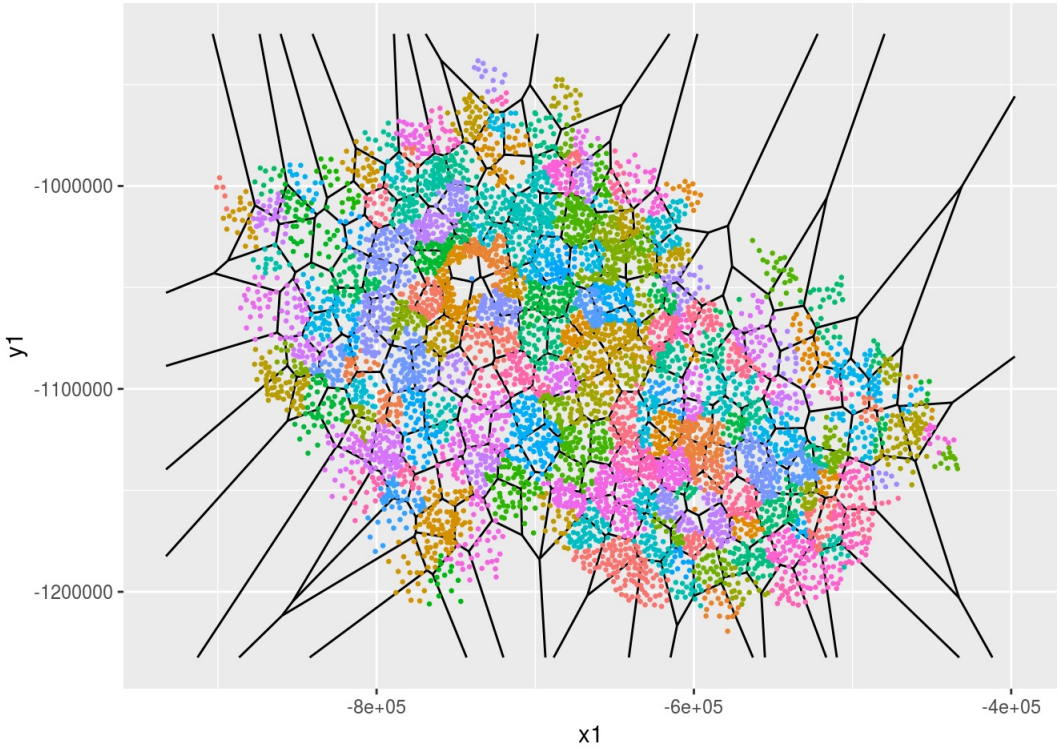
```
print(paste0('Number of clusters : '  ,  max(clusters)))
```

```
## [1] "Number of clusters : 202"
```

Voronoi map

HIDE

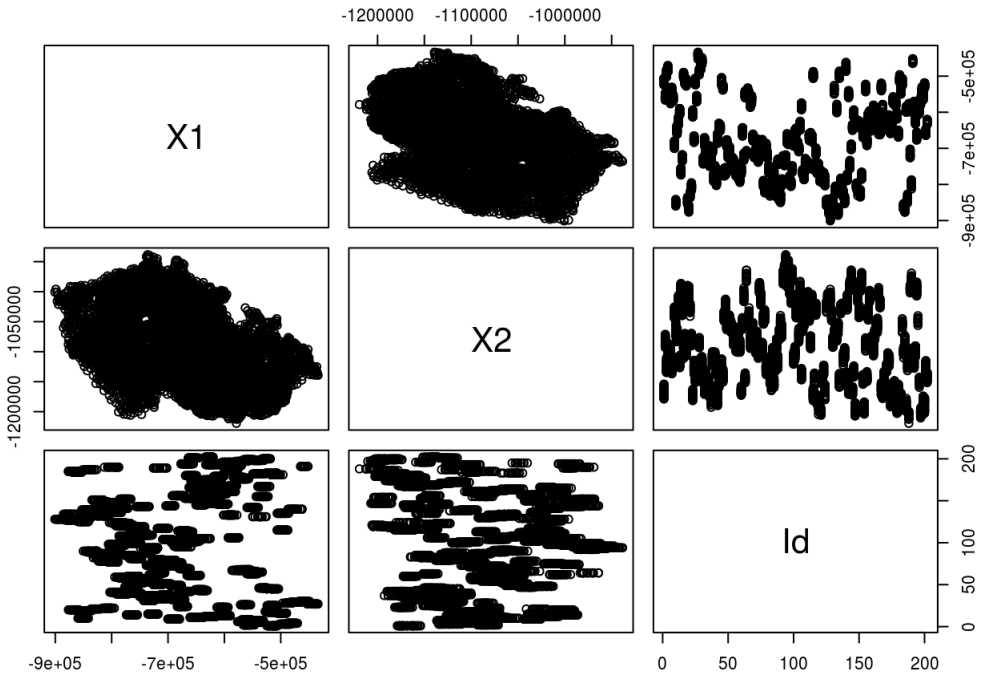
```
s = omega
s$order = clusters
q1 = s%>%
  group_by(order)%>%
  summarize(centroids = mean(x,na.rm= TRUE))
q2 = s%>%
  group_by(order)%>%
  summarize(centroids = mean(y,na.rm= TRUE))
q = inner_join(q1,q2,by='order')

vor = deldir(q$centroids.x,q$centroids.y)

R =
  ggplot() +
  geom_segment(
    aes(x = x1, y = y1, xend = x2, yend = y2),
    size = 0.5,
    data = vor$dirsgs,
    linetype = 1,
    alpha = 1 ) +geom_point(data=reg ,aes(x = reg$x, y = reg$y, group = reg$GraOrp, color=reg$GraOr
p),size=0.5)+theme(legend.position = "none")
R
```



CSV and statical comparison with old regions

HIDE

```
HH= tile.list(vor)
LL=cbind(reg$x,reg$y)
VV = NULL
for (i in seq(1, length(q$order))){
  KK = as.matrix(cbind(HH[[i]]$x,HH[[i]]$y) )
  MM = pnt.in.poly(LL,KK)
  FF = filter(MM,MM$pip==1)
  FF$pip = NULL
  FF$Id = HH[[i]]$ptNum
  VV = rbind(VV,FF)}

plot(VV)
```



HIDE

```
vv = na.omit(VV)
vv = unique(vv)
vv
```

1
2
4
5
6
7
8
9
10
11

1-10 of 6,257 rows | 1-1 of 4 columns

Previous **1** 2 Next

HIDE

```
kkk=(filter(lll,lll$boolean==TRUE))
correlation_lll = length(kkk$x)/length(lll$x)
print(paste0("city correlation = ", correlation_lll))
```

```
## [1] "city correlation = 0.234229828850856"
```

HIDE

```
g=NULL
g$cities=lll$`new regions`

h=NULL
h$cities=lll$region
compare(g,h)
```

```
## Component "cities": 4698 string mismatches
```

HIDE

```
# DENSITY-BASED CLUSTERING ----
#library(dbscan)
#kNNdistplot(omega, k = 3)
#abline(h = 0.7, col = "red", lty = 2)
#fitD <- dbscan(omega, minPts =1,eps=25000)
#fitD
#plot(omega, col = fitD$cluster)
```

Code of a enhanced X-means algorithm

HIDE

```
Lmin <- 10000
L <- list()
points = reg
points$ZobZkr = NULL
points$GraOrp = NULL
points$pix = NULL
points$lon = NULL
points$lat = NULL
points$cisdan = NULL
points = na.omit(points)
centroid = c(cisreg_cz0$x_cen,cisreg_cz0$y_cen)
##calculation of the largest distance between the centroid and the cities of the region newly create
d ----
distance_max = function(centroid,points){
  points$x1=centroid[1]
  points$y1=centroid[2]
  points$distance = sqrt((points$x-points$x1)^2+(points$y-points$y1)^2)

  return(max(points$distance))}
#Elbow trick to find the best k for the kmeans algorithm ----
best_K = function(points,kmax){
  k <- list()
  for(i in 1:kmax){
    k[[i]] <- kmeans(na.omit(points), i)
  }
  betweenss_totss <- list()
  for(i in 1:kmax){
    betweenss_totss[[i]] <- k[[i]]$betweenss/k[[i]]$totss
  }
  A = list()
  for (i in seq(1:(kmax-2))){
    A[[i]] = betweenss_totss[[i+2]]-2*betweenss_totss[[i+1]]+betweenss_totss[[i]]
    }
  return(which.min(A)+1)}

#creation of a recursive function that cluster the points taking the limitation of distance into co
nsideration ----

Clu <- function(centroid,points){
  if(distance_max(centroid,points)>Lmin){
    k=best_K(points,kmax =3)
    kms = kmeans(na.omit(points),k)
    voronoi = tile.list(deldir(as.data.frame(kms$centers)))
    all_points = cbind(reg$x,reg$y)
    for (i in seq(1:k)){
      K = as.matrix(cbind(voronoi[[i]]$x,voronoi[[i]]$y))
      M = pnt.in.poly(all_points,K)
      F = filter(M,M$pip==1)
      F$pip = NULL
      L =list.append(Clu(centroid = voronoi[[i]]$centers,points = F))
      }
    }
  else
  {return(L)}
}
```