USAGE OF R IN DEFINING LABOUR MARKET AREAS

PAWEŁ STOPIŃSKI

STATISTICAL OFFICE IN BYDGOSZCZ, POLAND

LABOUR MARKET AREA

A functional territory where people both live and work.

METHODS USED IN EUROPE

- LAM (Sweden)
- GEA (Spain)
- TTWA (Great Britain)

DATA SOURCES

- Population Census 2011
- administrative registers

CRUCIAL DEFINITIONS

- size= R_A
- self-containment=$\min\{t_{AA}/W_A, t_{AA}/R_A\}$

t_{AB} – number of people living in an area A and working in an area B

W_A – number of people working in an area A

R_A – number of employed residents of an area A

INPUT PARAMETERS

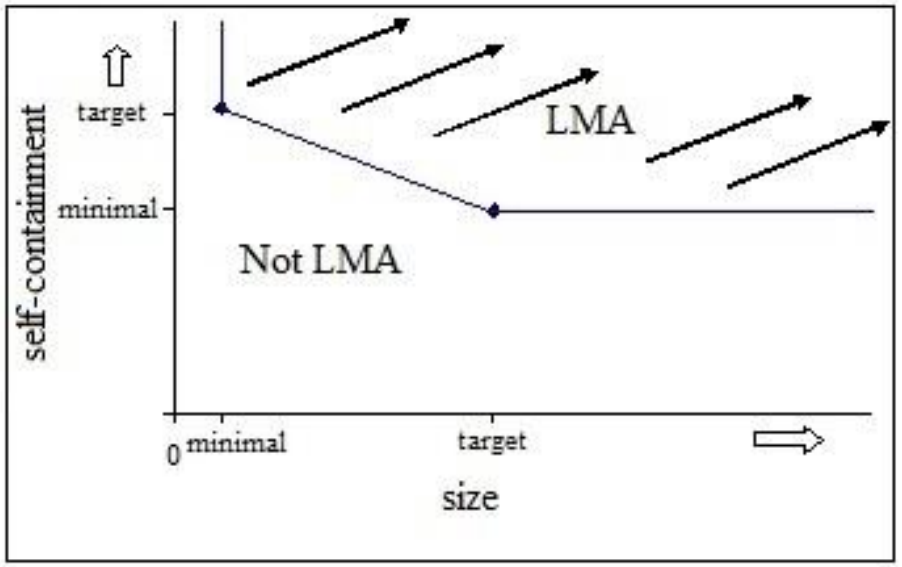
- minimal size ($size_{min}$)
- target size ($size_{target}$)
- minimal self-containment (sc_{min})
- target self-contaiment (sc_{target})

CRITERION OF BEING A VALID LMA

based on X-equation

- $\min\{sc_A, sc_{target}\}/sc_{target} * (1-(1-sc_{min}/sc_{target}) * \max\{(size_{target}-size_A)/(size_{target}-size_{min}), 0\}) >= size_{min}/size_{target}$

CRITERION OF BEING A VALID LMA



AUTHORS OF THE R SCRIPT

- Michele d'Alo (ISTAT, Italy)
- Luisa Franconi (ISTAT, Italy)
- Guido van den Heuvel (ONS, The Netherlands)

INPUT FILE - STRUCTURE

- community_live
- community_work
- amount

LWCLUS - STRUCTURE

- cluster_live
- cluster_work
- commuters

MARGINALS - STRUCTURE

- community
- cluster
- EMP_live

CLUSTERLIST - STRUCTURE

- cluster
- EMP_live
- EMP_work
- validity
- EMP_live_work
- SC_demand_side
- SC_supply_side

CALCULATING X-EQUATION FOR EACH CLUSTER

```
>LWSelf <- LWClus[LWClus$cluster_live == LWClus$cluster_work, c("cluster_live", "amount")]
>LWSelf <- merge(marginals, LWSelf, by.x = "cluster", by.y = "cluster_live",all.x=T)
>LWSelf$msc<- vectorMinFULL (LWSelf$amount / LWSelf$amount_live, LWSelf$amount /LWSelf$amount_work)
>LWSelf$Y<- LWSelf$msc/tarSC
>LWSelf$Z <- 1 - (1 - (minSC / tarSC)) * LWSelf$sizeMeasure
>LWSelf$validity <-  (LWSelf$Y * LWSelf$Z * (target self-containment/ minimal self-containment))
>minValidity <- min(LWSelf$validity,na.rm=T)
>validity.new<-list(LWSelf[LWSelf$validity == minValidity, c("cluster", "validity")][1,], LWSelf)
```

CHECKING COHESION BETWEEN THE CLUSTERS

The cluster with the least value of the validity function is joined with the cluster, which it has the strongest cohesion with.

$$cohesion_{AB}=t_{AB}/R_A*t_{AB}/W_B+t_{BA}/R_B*t_{BA}/W_A$$

COHESION IN THE R SCRIPT

```
>cohesion <- LWClus[LWClus$cluster_live != LWClus$cluster_work, ]

>cohesion <- merge(cohesion, marginals[, c("cluster", "amount_live")], by.x = "cluster_live", by.y = "cluster")

>cohesion <- merge(cohesion, marginals[, c("cluster", "amount_work")], by.x = "cluster_work", by.y = "cluster")

>cohesion$strengthOnesided <- cohesion$amount / cohesion$amount_live * cohesion$amount / cohesion$amount_work

>aggregate(data.frame(strength = cohesion$strengthOnesided), list(cluster1 = cohesion$cluster_live, cluster2 = cohesion$cluster_work), FUN = sum)

>cohesion <- cohesion[order(cohesion$cluster1, cohesion$strength),]

>cohesion <- cohesion[!duplicated(cohesion$cluster1, fromLast = TRUE), c("cluster1", "cluster2")]
```

DISSOLVING CLUSTERS

- X-equation is recalculated for new clusters.
- If the lowest validity does not decrease, the clusters stay joined and the algorithm starts from the beginning.
- If it decreases, the cluster with the lowest validity should be dissolved. Communities from the dissolved cluster need to be regrouped.

REGROUPING CLUSTERS

- The communities in the cluster are sorted decreasing by number of workers living outside the community plus number of residents of the community minus number of people living and working in the community.
- The first community is assigned to the cluster it has the strongest cohesion with.

REGROUPING CLUSTERS

- If the validity increases, first community is assigned to the most linked cluster and the other communities from dissolved cluster are regrouped into another one.
- If the validity decreases, first community is assigned to zero cluster and each community of dissolved cluster is assigned individually to the cluster, it has the strongest cohesion with.

NUMBER OF COMMUNITIES IN ‚ZERO CLUSTER'

size		self-containment		number of communities in 'zero cluster'
minimal	target	minimal	target	
3000	4000	0.6	0.7	2765
3000	5000	0.2	0.3	64
3000	5000	0.2	0.8	836
3000	5000	0.2	0.9	880
3000	5000	0.3	0.8	867
3000	5000	0.3	0.9	909
3000	5000	0.4	0.8	1186
3000	5000	0.4	0.9	1316
3000	5000	0.5	0.6	1859
3000	5000	0.5	0.8	1956
3000	5000	0.6	0.7	2758
3000	5000	0.6	0.8	2791
3000	5000	0.6	0.9	2720
10000	15000	0.6	0.7	2618

THE LAST STAGE OF THE ALGORITHM

Communities being a part of ’zero-cluster’ joined one by one to clusters which they have the strongest connection with.

‚FINDCLUSTERS' FUNCTION

```
findClusters <- function(filename, inDir, minSZ,minSC,tarSZ,tarSC,
verbose,sink.output,file.codici0,C0assign.name,ComNotAssigned.name)
```

- filename – name of the input file
- inDir – name of the input directory
- minSZ – minimal size
- minSC – minimal self-containment
- tarSZ – target size
- tarSC – target self-containment

‚FINDCLUSTERS' FUNCTION

- verbose – boolean parameter allowing to print on the screen the results after each stage
- sink.output – file containing the situation before dissolving ‚zero cluster'
- file.codici0 – file containing communities with no residents
- C0.assign.name – file containing the details of communities in ‚zero cluster'
- ComNotAssigned.name – file containing communities not assigned to any Labour Market Area

SAVING THE RESULTS

```
>write.table(out[[1]]$marginals, file =
paste(outDir,paste("R1.0",filename,"marginals",minSZ,minSC,tarSZ,tarSC,".csv",s
ep="_"), sep="\\"), sep = ";", col.names = TRUE,row.names = FALSE,append=F)

>write.table(out[[1]]$clusterList, file = paste(outDir,
paste("R1.0",filename,"clusterList",minSZ,minSC,tarSZ,tarSC,".csv",sep="_"),
sep="\\"), sep = ";", col.names = TRUE,row.names = FALSE,append=F)

>write.table(out[[1]]$LWClus, file = paste(outDir,
paste("R1.0",filename,"LWClus",minSZ,minSC,tarSZ,tarSC,".csv",sep="_"),
sep="\\"), sep = ";", col.names = TRUE,row.names = FALSE,append=F)
```

DRAWING A MAP – USED PACKAGES

- maps
- sp
- maptools
- mapdata

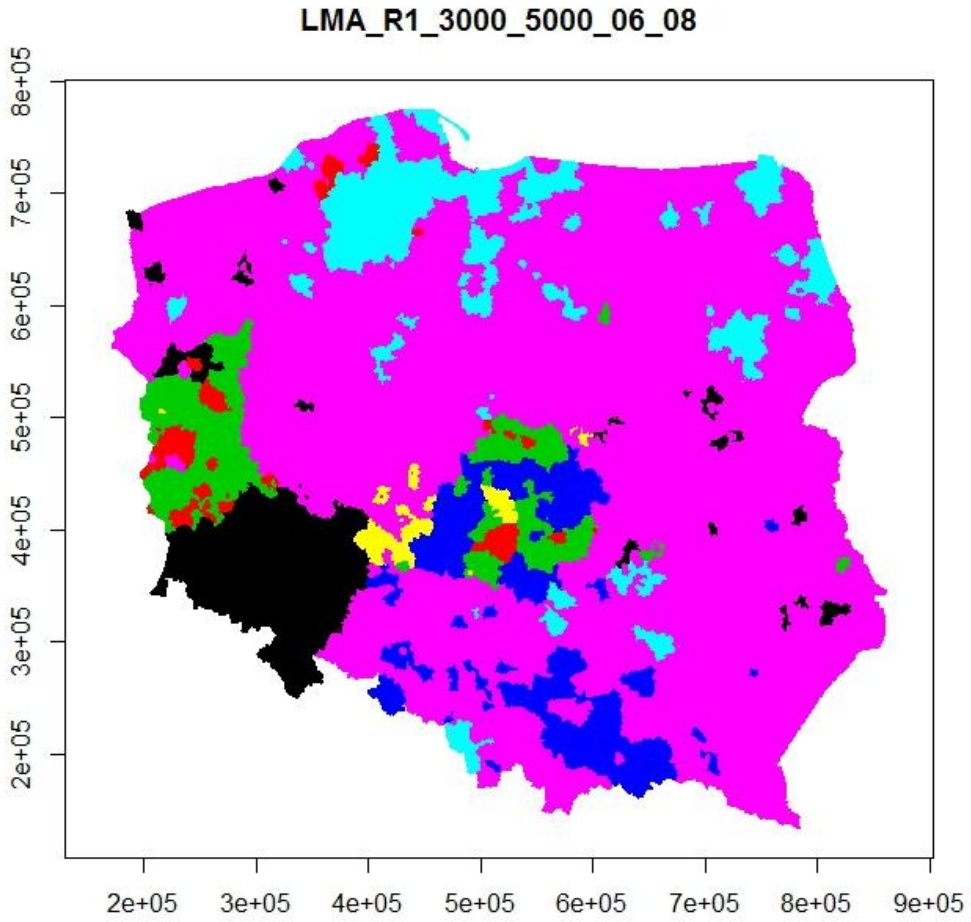
DRAWING A MAP

```
>shape<-readShapeSpatial(paste(inDir, shp.file,sep="\\"))
>shape@data<-cbind(shape, out[[2]]$clusterList)
>class <- as.factor(shape@data$LMA)
>plot(shape, border = class, col=class)
```

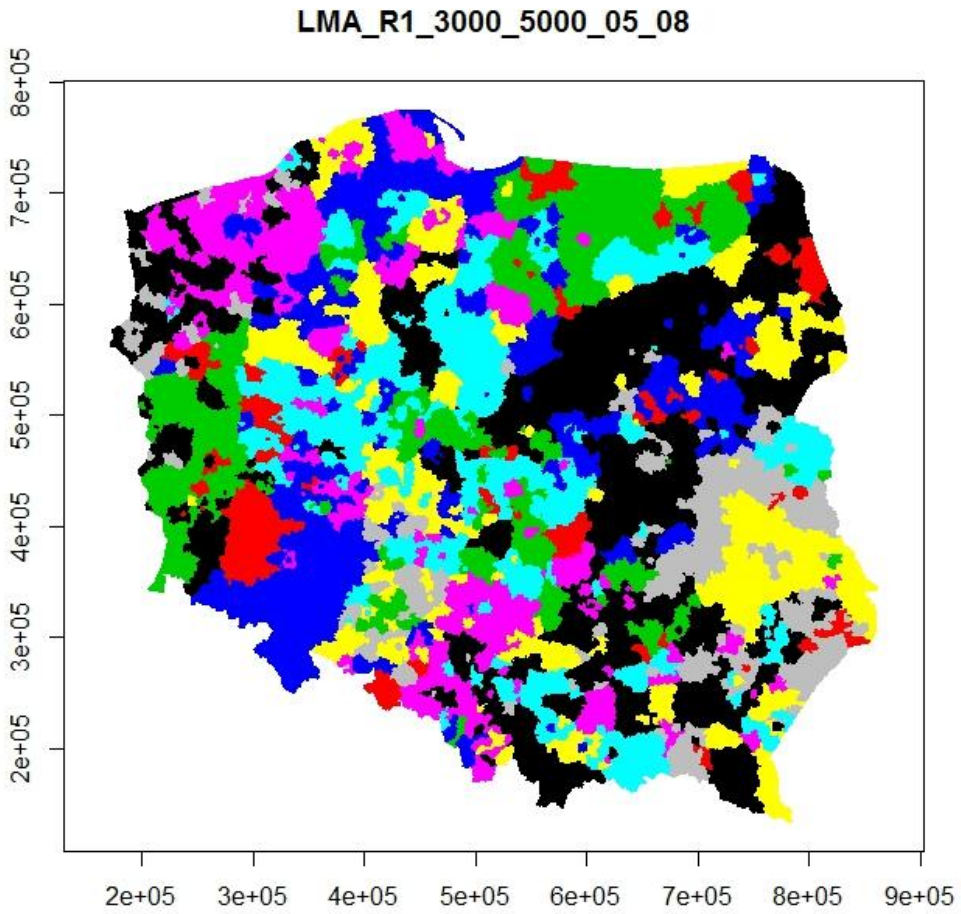
NUMBER OF LABOUR MARKET AREAS

size		self-containment		number of Labour Market Areas
minimal	target	minimal	target	
3000	4000	0.6	0.7	7
3000	5000	0.2	0.3	389
3000	5000	0.2	0.8	329
3000	5000	0.2	0.9	324
3000	5000	0.3	0.8	256
3000	5000	0.3	0.9	254
3000	5000	0.4	0.8	152
3000	5000	0.4	0.9	140
3000	5000	0.5	0.6	46
3000	5000	0.5	0.8	47
3000	5000	0.6	0.7	7
3000	5000	0.6	0.8	7
3000	5000	0.6	0.9	8
10000	15000	0.6	0.7	8

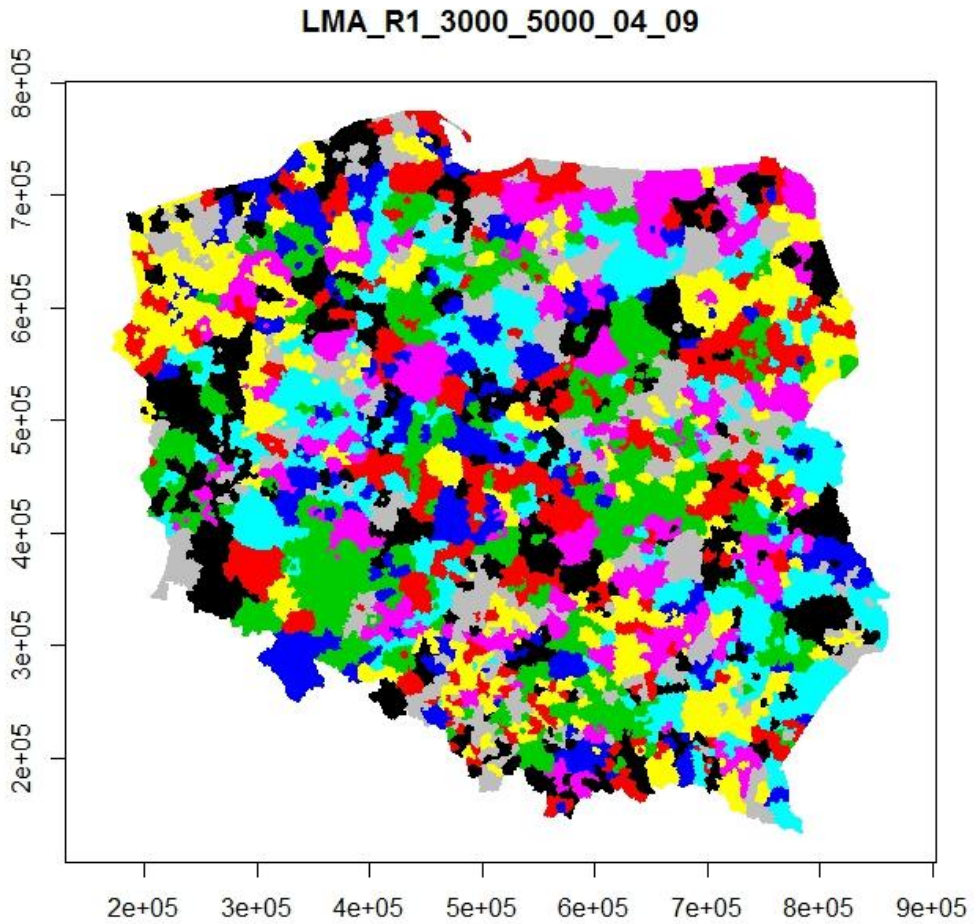
RESULTS – MAP 1



RESULTS – MAP 2



RESULTS – MAP 3



Thank you for your attention.