Usage Of R in Defining Labour Market Areas

Pawel STOPINSKI (p.stopinski@stat.gov.pl)
Statistical Office in Bydgoszcz, Poland

ABSTRACT

Labour Market Area is a territory in which high rate of people both live and work. It does not need to be consistent with area restricted by administrative boarders. It seems rather obvious that administrative boarders are not always a proper criterion in making certain decisions. The government should sometimes base on functional regions instead. That is the reason for which in 2013 Eurostat decided to from the Task Force responsible of creating a common methodology of defining Labour Market Areas. A couple of member states have already had certain experience in using their own definitions of Labour Market Areas. Nevertheless, due to independence of each method the results achieved in different countries were obviously incomparable. This caused a need to create a methodology, which would be universal for entire European Union. According to proposal there are two criteria deciding whether an area is a Labour Market Area or not – size (number of employed inhabitants) and self-containment, which is a minimal value of the following: 1) the proportion of an area's employed population that works within the area and 2) the proportion of jobs within an area that are filled by residents of that area. Since the results should be possibly stable, population censuses are desirable sources. As travel-to-work matrices contain relations between places of living and places of work at LAU-2 level, datasets may have large size (for Poland almost 350 000 records). The aim is to join areas into clusters so that all of them fulfill conditions to be considered as Labour Market Area. In each step only two areas are joined. Then computations for all new areas need to be performed from the beginning. There are also certain situations when once joined areas are divided, which makes the whole process more complicated. R seems to be a proper tool to carry out necessary analyses of such a big dataset.

Keywords: *Labour Market Area, R Project, self-containment, size, X-equation*
JEL Classification: *Z*

INTRODUCTION

Administrative boundaries stem from geographical, historical and other conditions. Nevertheless they are not always a proper criterion in making decisions by governors. That is why the role of functional regions in official statistics should keep increasing.

A problem of defining certain functional regions has been noticed in the previous century. As Standard Metropolitan Areas where defined in the USA already in 1949 (Frey and Speare, 1992). In 2002 OECD collected information about the definitions of functional regions from 22 member countries. It turned out that 17 of them defined functional regions in terms of labour market. (Cattan, 2002)

Labour Market Area is an example of functional region. It is an appropriately big territory where certain (preferably high) rate of people lives and travels to work within this area. Some European countries started taking in consider this problem several years ago, but their ways of dividing their territories in Labour Market Area are often completely different. For instance Swedish method LAM (Lokala arbetsmarknader) assumes choosing centers and building Labour Market Areas around them and Spanish method GEA is non-deterministic and one can obtain different results from the same data. In Poland some research was also made. Statistical Office in Poznan analyzed travel to work in 2010 and Statistical Office in Bydgoszcz described delimitation of functional region in Kujawsko-pomorskie voivodship.

A common way of defining Labour Market Areas for entire European Union could help in comparing regions in various Member States and make more equitable decisions. For this reason Eurostat decided to create a Task Force responsible for unifying methods used in different countries and creating one common methodology. The Task Force has not finished its work yet and did not create any official document containing results. Nevertheless, some suggestions have been made and an application prepared in R software is the crucial part of the whole project.

Method suggested by the Task Force is based on British methodology TTWA (Travel-To-Work Areas). It was already considered as the most proper one in 1990s – "*the TTWA method had been shown to be the 'best practice' for defining local labour market areas across Europe and so it was the model on which the Eurostat guidelines for their employment zone definitions were based.*" (Coombes, 2000). It requires considering two parameters – size and self-containment. Size is the number of economically active people living in certain area. Self-containment is the lower one of two values: first is the number of people both living and working in the area divided by the number of people living there and second is the number of people both living and working in the area divided by the number of people working there. Both parameters need to be relatively high. An area which will not fulfill required conditions will not be considered as a valid Labour Market Area.

Initially every LAU-2 area is the potential Labour Market Area. For each a number of people living in one of them and working in one another is necessary. In order to make Labour Market Areas stable the most preferred sources of data are population censuses, since in the most countries it is carried out once in ten years. Nevertheless also data based on administrative registers is acceptable. In Poland Population Census 2011 was based on two sources – data from surveys and administrative registers. For Labour Market Area purposes data from administrative part of census is taken.

R software allows to load databases, carry out necessary computations and obtain the way the country is divided into Labour Market Areas when chosen values of parameters are used. The last stage of the process is drawing an appropriate map with marked Labour Market Areas.

METHODOLOGY

As of the end of February 2015 work is still in progress. General idea is clear, but questions concerning some minor methodological details remain open. At this

stage Task Force Members are supposed to test their data with R program prepared by Michele d'Alo and Luisa Franconi from the Italian National Institute of Statistics (ISTAT) in cooperation with Guido van den Heuvel from the Office of Statistics Netherlands (ONS).

The input file used in the program should consist of three columns – *community_live* (code of community of living), *community_work* (code of community of working) and amount (number of people living in one community and working in another one). After having sorted the input file by *community_live* another one should be created. It should contain the code of each community and number of people living in it. If the input file is called *LWCom*, then in order to create the file with number of residents a following command can be used:

```
>residents<-aggregate(LWCom$amount,by=list(LWCom$communi
ty_live),sum)
```

The file should contain two columns called "code" and "residents".

According to this file in Poland there are 3081 communities. In fact there are more, since Warsaw is divided into several districts, but those are undistinguishable in administrative register, which was the main source for the data, so the entire capital city needs to be considered as one municipality. This entails that Warsaw is the community with the biggest number of working inhabitants (541 419 people). The one with the smallest number of working inhabitants is a rural part of Krynki – a community in the northern east of Poland (67 people).

Since in some countries there are communities without any people living in them, a vector including their codes should be created. Then they should be deleted from the list of communities in all files.

```
>codes.0<-sort(unique(residents$Code[residents$residents==0]))
>write.table(codes.0,file=file_with_0_residents,append=F,row.
names=F,col.names=T,sep=";",quote=F)
>LWCom<-LWCom[!(LWCom$community_live%in%codes.0),]
```

In order to make a user aware of current state of division some functions creating specific files are attached. Thanks to *InitClusterList* function user obtains an object *clusterList* consisting of three columns: *community*, *cluster* and *residents*. *InitLWClusterFULL* creates an object *LWClus* similar to the input one, but instead of communities there are clusters. *InitMarginalsFULL* provides another one – *marginals* – with clusters, number of people living and number of people working in them. Function *initCLusterFULL* contains three functions described above.

While data frame containing number of people travelling to work from one community to another changes, so in order to have an original relation between communities the following frame is created:

```
>CommunityWorkersOrig<-merge(clusterData$clusterList,clusterDa
ta$marginals,by="cluster")[,c("community","amount_work")].
```

A frame containing information about number of people living and working in the same community is also defined:

```
>Community_live_work<-
LWCom[LWCom$community_live==LWCom$community_work,c("community_
live","amount")]
>names(Community_live_work) <-c("community","live_work")
```

A criterion deciding whether a cluster *A* can be considered as a valid Labour Market Area is X-equation:

*X = Y * Z*, with

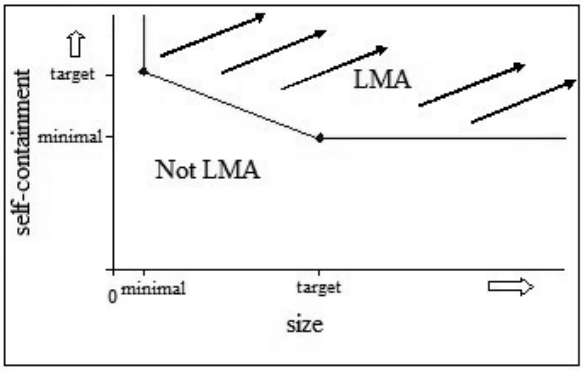
Y = (1 / target_self_containment) min(A_self_containment, target_self_containment),

Z = 1 - (1-minimal_self_containment / target_self_containment) max(target_size – A_size) / (target_size – minimal_size), 0)*.

The chart of X-equation is presented below.

Criteria of being a valid LMA

Figure 1



Areas which fulfill the condition to have a value calculated from X-equation above the line are valid Labour Market Areas. In other words, when

X>=minimal_size/target_size,

an area can be concerned as valid Labour Market Area.

For the beginning each community is concerned as potential Labour Market Area. Of course after having X-equation computed it probably will not turn out that value for each one is above the line. Then there is going to be a need to join some of communities.

The decision which clusters should be aggregated is made on the basis of self-containment. The one which has the lowest value of X-equation (called "least self-contained" in the program) is supposed to be regrouped.

For computations concerning the least self-contained cluster communities with no people living or no people working should be excluded.

For communities left there is a need to focus on the number of people living and working in the same community by creating a vector *LWSelf*.

```
>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)
```

As a result a list of two components is obtained – first one is the cluster with its validity and the second one is the list of all clusters and it consists of columns: *cluster*, *amount_live*, *amount_work*, *amount*, *msc*, *Y*, *sizeMeasure*, *Z* and *validity*, where *size_measure* is a part of factor *Z* and it is calculated as
(target size - LWSelf$amount_live) / (target size – minimal size)

Choice of cluster which the least self-contained cluster should be put together with is made after having calculated cohesion.

Cohesion between cluster *A* and cluster *B* is defined as:

$$cohesion_{AB}=t_{AB}/R_A*t_{AB}/W_B+t_{BA}/R_B*t_{BA}/W_A \quad [1]$$

where

t_{AB} – number of people living in cluster *A* and working in cluster *B*,
W_A – number of people working in cluster *A*,
R_A – number of people living in cluster *A*,
t_{BA} – number of people living in cluster *B* and working in cluster *A*,
W_B – number of people working in cluster *B*,
R_B – number of people living in cluster *B*.

In R program cohesion is determined in the following way:

```
>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")
```

After eliminating clusters with 0 amount of people working or people living a column *strengthOnesided* is added and calculated as:

```
>cohesion$strengthOnesided <- cohesion$amount / cohesion$amount_
live * cohesion$amount / cohesion$amount_work,
```

which is the first factor of eqn [1].

The second factor is obtained by swapping columns where *cluster_live>cluster_work* and adding two factors together:

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

cluster 2 which has the strongest cohesion with cluster 1 is chosen. Following two commands are performed:

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

Then the least self-contained cluster is joined with the cluster which it has the strongest cohesion with.

The X-equation value of the cluster after connection should be counted once again. If it decreases the community with the least validity should be put on so-called "reserve list". It is still not decided whether the community should be put on the list in every case or just when after recalculating X-equation it turns out that the community stopped being valid Labour Market Area. In the discussed program the community is put on the reserve list no matter if after recalculating X-equation it fulfills conditions of being a valid Labour Market Area or not.

```
>LWCom<-rbind(LWCom,rep(fict.community,ncol(LWCom)))
>LWCom$amount[LWCom$community_live==fict.community]<-0
>residents<-rbind(residents,c(fict.community,0))
>clusterData <- initClusterFULL(LWCom, residents)
```

During dissolving a cluster communities receive temporary identity numbers with negative values (in order to emphasize their temporality) and then function *initClusterFULL* is performed for new structure of clusters.

Communities from the dissolved cluster need to be regrouped. If there is a need to regroup more than one community (for example when a cluster dissolved consisting of two or more communities), a criterion of sorting should be prepared. The communities are sorted decreasing by number of workers living outside the cluster plus number of residents of the cluster minus number of people living and working in the cluster.

Clusters with negative identity numbers are chosen from the list:

```
>LWClusDissolve <- clusterData$LWClus[xor(clusterData$LWClus$clus
ter_live < 0, clusterData$LWClus$cluster_work < 0),]
```

and rows which have zero number of people living or working in the community are deleted. Clusters which still left need to be sorted by the number of

the cluster and strength of the cohesion and merged again.

After that cluster needs to be sorted by validity and compared with the old cluster called *leastSelfContained*. If all validities are bigger the cluster can be changed for the new one.

```
>if(all(validity.new[[2]]$validity[validity.
new[[2]]$cluster>0]>=leastSelf
ContainedFULL[[2]]$validity[leastSelfContainedFULL[[2]]$cluster>
0])){clusterData=clusterData.new}
```

Otherwise the first community is assigned to the 'zero cluster':

```
>clusterData$clusterList$cluster[clusterData$clusterList$cluster=
=index.com.2diss]=0
>clusterData$LWClus$cluster_live[clusterData$LWClus$cluster_
live==index.com.2diss]=0
>clusterData$LWClus$cluster_work[clusterData$LWClus$cluster_
work==index.com.2diss]=0
>clusterData$marginals$cluster[clusterData$marginals$cluster==ind
ex.com.2diss]=0,
```

where

```
 >index.com.2diss=clusterData$clusterList$cluster[clusterData$clu
sterList$community==com.cluster2dissolve$community[1]]
```

Type of assignment of the first community plays and important role, since if it is assigned to another cluster (not the 'zero cluster') all other communities from the dissolved cluster are regrouped into one cluster. In the other case the cluster is totally dissolved and each community except of the first one is assigned to the cluster having the biggest value of cohesion with it.

If there is a need to regroup more than one community (for example when a cluster dissolved consisting of two or more communities), a criterion of sorting should be prepared. The communities are sorted decreasing by number of workers living outside the cluster plus number of residents of the cluster minus number of people living and working in the cluster.

Those steps need to be repeated as long as not all clusters fulfill conditions of being valid Labour Market Areas.

It is solved using standard function 'repeat' and

```
>if (leastSelfContained$validity==1)
{print("break for validity=1")
         break}
```

inside it.

As the last stage 'zero cluster' needs to be dissolved and communities being a part of it should be assigned to other clusters. Current state of division before assignment is saved as a vector

```
>clusterDataBeforeZeroCluster<-clusterData$clusterList[clusterDat
a$clusterList$community!=fict.community,]
```

Program allows to check, how many communities could not be assigned to "regular" clusters and it creates output files for division of the country taking into account 'zero cluster'.

Scale of this phenomenon for Poland is shown in Table 1.

Number of communities in 'zero cluster' with different values of parameters used

Table 1

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

If the is more than one community in 'zero cluster', they need to be ordered by number of workers living outside plus number of residents minus number of people living and working in the community.

All communities are joined one by one to clusters which they have the strongest connection with.

All functions described above are collected into one general called findClusters working on following arguments: filename, input directory, minimal size, minimal self-containment, target size, target self-containment, argument allowing or not to print intermediate results on the screen, localization of output directory, file containing communities with 0 residents, file with assignments to cluster 0 and file with communities which were not able to be attached to any Labour Market Area. The only thing user needs to do is to define those values and run this function.

RESULTS

As a result of the procedures performed above a list of two components is obtained. The first one is the final result of the procedure. Results are saved in output directory in three files with '.csv' format:

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

Nevertheless, the most important part of the result is a data frame having an 'Rdata' format and containing all the result of the procedure, which is also saved in the output directory.

```
>name_file.output<-paste("R1.0",filename,minSZ,minSC,tarSZ,tarSC,".
Rdata",sep="_")
>save(out[[1]],file=paste(outDir, name_file.output, sep="\\"))
```

This element is supposed to be used in the next stage. The others are created for informational purposes.

The second component of the procedure result concerns only communities assigned to Labour Market Areas without those put on reserve list. It also consists of analogous three elements, which are saved to output files and naturally a data frame in '.Rdata' format. Names of the files contain *before ZeroClusterAssignment*. Importance of this components stems from the fact that it may be helpful in analyses of the results while it shows a scale of attaching municipalities from 'zero clusters' to valid Labour Market Areas defined in the way above.

The last stage of the procedure is drawing a map of the country divided into Labour Market Areas. In order to enable user to obtain such a map using R, four additional packages ('maps', 'sp', 'maptools' and 'mapdata') are needed to be installed.

The map should base on an '.shp' file containing communities with the same codes as the ones used in the entrance file, shape length and shape area. Function *readShapeSpatial* enables reading a file in order to prepare a plot of the country and its municipalities. An object defined this way is called *shape*.

Next an additional frame is joined with shape as follows:

```
>shape@data<-cbind(shape, out[[2]]$clusterList)
```

Finally, a colored map of the country divided into Labour Market Areas defined in the way above is created.

```
>plot(shape, border = class, col=class)
```

where *class* is defined as:

```
>class <- as.factor(shape@data$LMA)
```

As per the end of February 2015 the program was run several times with sets of different parameter values (so far mainly minimal and target self-containment values were changed). The results are shown in Table 2.

Number of Labour Market Areas with different values of parameters used

Table 2

size		self-containment		number of Labour
minimal	target	minimal	target	Market Areas
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	0.4	0.8	46
3000	5000	0.4	0.9	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

CONCLUSIONS

Task Force has not finished its work yet and further analyses need to be performed in order to choose optimal parameter values for Labour Market Areas.

R application allows to analyze big datasets and obtain results in reasonable time. Running an algorithm for 3081 communities in Poland lasts about 12 hours. Within this time user gets a map of the country divided into Labour Market Areas in the way they were defined above. However, the script still going to be improved in order to shorten the time of performance, since in some bigger countries (e.g. Great Britain) program is not efficient enough.

The program prepared by Michele d'Alo, Luisa Franconi in cooperation with Guido van den Heuvel gives an opportunity to check current state of division at each stage of the procedure, which allows to understand better the process of creating a division into Labour Market Areas.

As long as ultimate decisions about all details of Labour Market Areas defining are not set, some further modifications of the program seem necessary. The aim for the future is to introduce a common method of defining Labour Market Areas for the entire European Union and the R program will probably become one of the most important tools used to achieve it.

REFERENCES

1. URL: http://www.istat.it/it/files/2014/12/Final-Report_LMA-v1-0-17102012.pdf - access on 20.02.2015
2. URL: www.istat.it/it/files/2014/12/final_TTWA_report.doc - access on 20.02.2015
3. URL: http://cran.r-project.org/web/packages/maptools/maptools.pdf - access on 05.03.2015
4. URL: http://bydgoszcz.stat.gov.pl/opracowania-biezace/opracowania-sygnalne/inne-opracowania/delimitacja-obszarow-funkcyjnych-wybranych-miast-w-wojewodztwie-kujawsko-pomorskim-czesc-1,5,4.html# - access on 05.03.2015
5. Cattan N (2002) Redefining territories: functional regions, DT/TDPC/TI(2002)3, Paris
6. Coombes M (2000), Geographic information systems: a challenge for statistical agencies, Research in Official Statistics, Volume 3, Number 2, pp. 77-87
7. Frey W and Speare A (1992), Metropolitan areas as functional communities: a proposal for a new definition, US Bureau of The Census, Population Studies Center, Michigan
8. Urząd Statystyczny w Poznaniu (2010), Dojazdy do pracy w Polsce, Zakład Wydawnictw Statystycznych, Poznań