**# miary cluster-based**

**# mapa Polski**

library(spdep)

library(rgdal)

library(maptools)

library(sp)

library(rgeos)

# mapy

setwd("E:/My all/&Naukowe/moje naukowe w toku/GRANT ERSA/# obliczenia SPAG/Lubelskie")

woj<-readOGR(".", "wojewodztwa") # 16 jedn.

pow<-readOGR(".", "powiaty") # 380 jedn.

projekcja<-"+proj=longlat +datum=WGS84"

woj <- spTransform(woj, CRS(projekcja))

pow <- spTransform(pow, CRS(projekcja))

# dane nt. struktury podziału terytorialnego powiaty-podregiony-województwa (dane powiatowe)

dane.pow<-read.csv("E:/My all/&Naukowe/moje naukowe w toku/GRANT ERSA/# obliczenia SPAG/Lubelskie/dane\_powiatowe\_2016.csv", sep=";", dec=",", header=TRUE)

dim(dane.pow)

dane.pow<-dane.pow[1:380, ]

**##############################\**

**# defining the data - lubelskie**

setwd("E:/My all/&Naukowe/moje naukowe w toku/GRANT ERSA/# obliczenia cluster-based")

dane<-read.csv("pivot lubelskie sekcje.csv", header=TRUE, dec=",", sep=";")

sum.by.regions<-dane[20, 2:6]

sum.by.sectors<-dane[1:19, 6]

sectors<-dane[1:19,1]

empl<-dane[1:19, 2:5]

m<-4 # numer of regions

n<-19 # numer of sectors

**# defining the data on firms**

# distribution of firms by size

firmy<-read.csv("pivot lubelskie firmy.csv", header=TRUE, dec=",", sep=";")

dim(firmy)

firmy<-firmy[1:19,]

**# matrix of size of companies**

size<-matrix(0, nrow=dim(firmy)[1]-1, ncol=dim(firmy)[2]-2)

size[,1]<-rep(5, times= dim(firmy)[1]-1)

size[,2]<-rep(30, times= dim(firmy)[1]-1)

size[,3]<-rep(150, times= dim(firmy)[1]-1)

size[,4]<-rep(600, times= dim(firmy)[1]-1)

size[,5]<-rep(1500, times= dim(firmy)[1]-1)

size

**##########**

**# defining the data - lubelskie**

setwd("E:/My all/&Naukowe/moje naukowe w toku/GRANT ERSA/# obliczenia cluster-based")

dane<-read.csv("pivot lubelskie PKD2.csv", header=TRUE, dec=",", sep=";")

sum.by.regions<-dane[86, 2:6]

sum.by.sectors<-dane[1:85, 6]

sectors<-dane[1:85,1]

empl<-dane[1:85, 2:5]

m<-4 # numer of regions

n<-85 # numer of sectors

**# defining the data on firms**

# distribution of firms by size

firmy<-read.csv("pivot lubelskie firmy PKD.csv", header=TRUE, dec=",", sep=";")

dim(firmy)

firmy<-firmy[1:85,] # bez sumy

**# matrix of size of companies**

size<-matrix(0, nrow=dim(firmy)[1], ncol=dim(firmy)[2]-2)

size[,1]<-rep(5, times= dim(firmy)[1])

size[,2]<-rep(30, times= dim(firmy)[1])

size[,3]<-rep(150, times= dim(firmy)[1])

size[,4]<-rep(600, times= dim(firmy)[1])

size[,5]<-rep(1500, times= dim(firmy)[1])

size

**##########**

**# wybór województwa z mapy województw i powiatowej**

woj.df<-as.data.frame(woj)

region<-woj[woj.df$jpt\_nazwa\_=="lubelskie",]

region<-spTransform(region, CRS(projekcja))

pow.reg<-pow[dane.pow$województwo=="Lubelskie",]

pow.reg<-spTransform(pow.reg, CRS(projekcja))

# wybór województwa dane

dane.reg<-dane.pow[dane.pow$województwo=="Lubelskie", ]

# tworzenie mapy podregionów z mapy powiatowej

map<-unionSpatialPolygons(pow.reg, dane.reg$podregion..układ.72.)

map<-spTransform(map, CRS(projekcja))

podreg.names<-row.names(map)

crds<-coordinates(map)

**# defining the spatial weights matrix**

cont.nb<-poly2nb(as(map, "SpatialPolygons"))

cont.listw<-nb2listw(cont.nb, style="W")

**# defining the distance between regions - przykładowe**

dist<-dist(crds, upper=TRUE)

dist

**##############################\**

**# defining the data - wielkopolskie**

setwd("E:/My all/&Naukowe/moje naukowe w toku/GRANT ERSA/# obliczenia cluster-based")

dane<-read.csv("pivot wielkopolskie sekcje.csv", header=TRUE, dec=",", sep=";")

sum.by.regions<-dane[20, 2:8]

sum.by.sectors<-dane[1:19, 8]

sectors<-dane[1:19,1]

empl<-dane[1:19, 2:7]

m<-6 # numer of regions

n<-19 # numer of sectors

**# defining the data on firms**

# distribution of firms by size

firmy<-read.csv("pivot wielkopolskie firmy.csv", header=TRUE, dec=",", sep=";")

dim(firmy)

firmy<-firmy[1:19,]

**# matrix of size of companies**

size<-matrix(0, nrow=dim(firmy)[1], ncol=dim(firmy)[2]-2)

size[,1]<-rep(5, times= dim(firmy)[1])

size[,2]<-rep(30, times= dim(firmy)[1])

size[,3]<-rep(150, times= dim(firmy)[1])

size[,4]<-rep(600, times= dim(firmy)[1])

size[,5]<-rep(1500, times= dim(firmy)[1])

size

**##########**

**# defining the data - śląskie**

setwd("E:/My all/&Naukowe/moje naukowe w toku/GRANT ERSA/# obliczenia cluster-based")

dane<-read.csv("pivot wielkopolskie PKD.csv", header=TRUE, dec=",", sep=";")

sum.by.regions<-dane[88, 2:8]

sum.by.sectors<-dane[1:87, 8]

sectors<-dane[1:87,1]

empl<-dane[1:87, 2:7]

m<-6 # numer of regions

n<-87 # numer of sectors

**# defining the data on firms**

# distribution of firms by size

firmy<-read.csv("pivot wielkopolskie firmy PKD.csv", header=TRUE, dec=",", sep=";")

dim(firmy)

firmy<-firmy[1:87,] # bez sumy

**# matrix of size of companies**

size<-matrix(0, nrow=dim(firmy)[1], ncol=dim(firmy)[2]-2)

size[,1]<-rep(5, times= dim(firmy)[1])

size[,2]<-rep(30, times= dim(firmy)[1])

size[,3]<-rep(150, times= dim(firmy)[1])

size[,4]<-rep(600, times= dim(firmy)[1])

size[,5]<-rep(1500, times= dim(firmy)[1])

size

**##########**

# wybór województwa z mapy województw

woj.df<-as.data.frame(woj)

region<-woj[woj.df$jpt\_nazwa\_=="wielkopolskie",]

region<-spTransform(region, CRS(projekcja))

# wybór województwa z mapy powiatowej

pow.reg<-pow[dane.pow$województwo=="Wielkopolskie",]

pow.reg<-spTransform(pow.reg, CRS(projekcja))

# wybór województwa dane

dane.reg<-dane.pow[dane.pow$województwo=="Wielkopolskie", ]

# tworzenie mapy podregionów z mapy powiatowej

map<-unionSpatialPolygons(pow.reg, dane.reg$podregion..układ.72.)

map<-spTransform(map, CRS(projekcja))

podreg.names<-row.names(map)

crds<-coordinates(map)

**# defining the spatial weights matrix**

cont.nb<-poly2nb(as(map, "SpatialPolygons"))

cont.listw<-nb2listw(cont.nb, style="W")

**# defining the distance between regions - przykładowe**

dist<-dist(crds, upper=TRUE)

dist

**##############################\**

**# defining the data - śląskie**

setwd("E:/My all/&Naukowe/moje naukowe w toku/GRANT ERSA/# obliczenia cluster-based")

dane<-read.csv("pivot śląskie sekcje.csv", header=TRUE, dec=",", sep=";")

sum.by.regions<-dane[20, 2:10]

sum.by.sectors<-dane[1:19, 10]

sectors<-dane[1:19,1]

empl<-dane[1:19, 2:9]

m<-8 # numer of regions

n<-19 # numer of sectors

**# defining the data on firms**

# distribution of firms by size

firmy<-read.csv("pivot śląskie firmy.csv", header=TRUE, dec=",", sep=";")

dim(firmy)

firmy<-firmy[1:19,]

**# matrix of size of companies**

size<-matrix(0, nrow=dim(firmy)[1]-1, ncol=dim(firmy)[2]-2)

size[,1]<-rep(5, times= dim(firmy)[1]-1)

size[,2]<-rep(30, times= dim(firmy)[1]-1)

size[,3]<-rep(150, times= dim(firmy)[1]-1)

size[,4]<-rep(600, times= dim(firmy)[1]-1)

size[,5]<-rep(1500, times= dim(firmy)[1]-1)

size

**##########**

**# defining the data - śląskie**

setwd("E:/My all/&Naukowe/moje naukowe w toku/GRANT ERSA/# obliczenia cluster-based")

dane<-read.csv("pivot śląskie PKD.csv", header=TRUE, dec=",", sep=";")

sum.by.regions<-dane[87, 2:10]

sum.by.sectors<-dane[1:86, 10]

sectors<-dane[1:86,1]

empl<-dane[1:86, 2:9]

m<-8 # numer of regions

n<-86 # numer of sectors

**# defining the data on firms**

# distribution of firms by size

firmy<-read.csv("pivot śląskie firmy PKD.csv", header=TRUE, dec=",", sep=";")

dim(firmy)

firmy<-firmy[1:86,] # bez sumy

**# matrix of size of companies**

size<-matrix(0, nrow=dim(firmy)[1], ncol=dim(firmy)[2]-2)

size[,1]<-rep(5, times= dim(firmy)[1])

size[,2]<-rep(30, times= dim(firmy)[1])

size[,3]<-rep(150, times= dim(firmy)[1])

size[,4]<-rep(600, times= dim(firmy)[1])

size[,5]<-rep(1500, times= dim(firmy)[1])

size

**##########**

# wybór województwa z mapy województw

woj.df<-as.data.frame(woj)

region<-woj[woj.df$jpt\_nazwa\_=="śląskie",]

region<-spTransform(region, CRS(projekcja))

# wybór województwa z mapy powiatowej

pow.reg<-pow[dane.pow$województwo=="Śląskie",]

pow.reg<-spTransform(pow.reg, CRS(projekcja))

# wybór województwa dane

dane.reg<-dane.pow[dane.pow$województwo=="Śląskie", ]

# tworzenie mapy podregionów z mapy powiatowej

map<-unionSpatialPolygons(pow.reg, dane.reg$podregion..układ.72.)

map<-spTransform(map, CRS(projekcja))

podreg.names<-row.names(map)

crds<-coordinates(map)

**# defining the spatial weights matrix**

cont.nb<-poly2nb(as(map, "SpatialPolygons"))

cont.listw<-nb2listw(cont.nb, style="W")

**# defining the distance between regions - przykładowe**

dist<-dist(crds, upper=TRUE)

dist

**##############################\**

**# defining the data - przykładowe**

setwd("E:/My all/&Naukowe/moje naukowe w toku/GRANT ERSA/# obliczenia cluster-based")

dane<-read.csv("example.csv", header=TRUE, dec=",", sep=";")

sum.by.regions<-dane[5, 2:8]

sum.by.sectors<-dane[1:4, 8]

sectors<-dane[1:4,1]

empl<-dane[1:4, 2:7]

m<-6 # numer of regions

n<-4 # numer of sectors

library(gtools)

**##############################\**

**# defining the data on firms**

# distribution of firms by size

firmy<-read.csv("firms.csv", header=TRUE, dec=",", sep=";")

dim(firmy)

**# matrix of size of companies**

size<-matrix(0, nrow=dim(firmy)[1], ncol=dim(firmy)[2]-2)

size[,1]<-rep(100, times= dim(firmy)[1])

size[,2]<-rep(50, times= dim(firmy)[1])

size[,3]<-rep(10, times= dim(firmy)[1])

size[,4]<-rep(1, times= dim(firmy)[1])

size

**##############################\**

**# defining the distance between regions - przykładowe**

dist<-read.csv("dist.csv", header=TRUE, dec=",", sep=";")

dim(dist)

colnames(dist)<-paste("region",1:m)

rownames(dist)<-paste("region",1:m)

dist

map<-readOGR(".", "mapa6wybr")

projekcja<-"+proj=longlat +datum=WGS84"

map<-spTransform(map, CRS(projekcja))

plot(map)

crdsb<-coordinates(map)

b<-1:6

text(crdsb[1:5,], labels=b[1:5])

text(15.7, 52.1, labels=b[6])

**# defining the spatial weights matrix**

cont.nb<-poly2nb(as(map, "SpatialPolygons"))

cont.listw<-nb2listw(cont.nb, style="W")

**##############################\**

**# wyniki zbiorczo**

wyniki.all<-matrix(0, ncol=m, nrow=13)

rownames(wyniki.all)<-c("Ogive\_index", "Krugman\_index", "National\_Averages\_Index\_(NAI)", "Hallet\_index", "Relative\_Diversity\_Index\_(RDI)", "Shannon’s\_H", "Relative\_H", "Theil’s\_H", "KLD", "Index\_of\_diversification", "Hachman", "RSI\_(max\_LQ)", "Gini")

colnames(wyniki.all)<-paste("region",1:m)

################################\

**# Ogive index**

wynik<-matrix(0, ncol=m, nrow=1)

colnames(wynik)<-paste("region",1:m)

rownames(wynik)<-c("Ogive index")

wynik

shares.in.columns<-matrix(0, ncol=m, nrow=n)

shares.in.columns

for(i in 1:n){

temp<-empl[i,]/sum.by.regions[1:m]

shares.in.columns[i,]<-as.matrix(temp[1,])}

shares.in.columns

colSums(shares.in.columns) # check of sums in columns

share\_star<-matrix(1/n, ncol=m, nrow=n)

diff.sq<-(shares.in.columns-share\_star)^2

diff.sq

for(j in 1:m){

wynik[1,j]<- sum(diff.sq[,j])/(1/n)}

wynik

wyniki.all[1,]<-wynik

wyniki.all

################################\

**# Krugman index, NAI, Hallet, Relative Diversity Index**

wynik<-matrix(0, ncol=m, nrow=4)

colnames(wynik)<-paste("region",1:m)

rownames(wynik)<-c("Krugman index", "National Averages Index (NAI)", "Hallet index", "Relative Diversity Index (RDI)")

wynik

shares.in.columns<-matrix(0, ncol=m, nrow=n)

shares.in.columns

for(i in 1:n){

temp<-empl[i,]/sum.by.regions[1:m]

shares.in.columns[i,]<-as.matrix(temp[1,])}

shares.in.columns

colSums(shares.in.columns) # check of sums in columns

share.in.columns.extra<-sum.by.sectors/sum(sum.by.sectors)

share.in.columns.extra

for(i in 1:m){

diff.abs<-abs(shares.in.columns[,i]- share.in.columns.extra)

wynik[1,i]<-sum(diff.abs)}

for(i in 1:m){

diff.abs<-(shares.in.columns[,i]- share.in.columns.extra)^2/share.in.columns.extra

wynik[2,i]<-sum(diff.abs)}

wynik[3,]<-wynik[1,]/2

wynik[4,]<-1/wynik[1,]

wynik

wyniki.all[2:5,]<-wynik

################################\

**# Entropy – Theil’s H, Shannon’s H, Relative H, KLD**

wynik<-matrix(0, ncol=m, nrow=4)

colnames(wynik)<-paste("region",1:m)

rownames(wynik)<-c("Shannon’s H", "Relative H", "Theil’s H", "KLD")

wynik

shares.in.columns<-matrix(0, ncol=m, nrow=n)

shares.in.columns

for(i in 1:n){

temp<-empl[i,]/sum.by.regions[1:m]

shares.in.columns[i,]<-as.matrix(temp[1,])}

shares.in.columns

colSums(shares.in.columns) # check of sums in columns

logs.matrix<-matrix(0, ncol=m, nrow=n)

logs.matrix<-shares.in.columns\*log(shares.in.columns)

logs.matrix

sum.columns<-colSums(logs.matrix)

sum.columns

share.in.columns.extra<-sum.by.sectors/sum(sum.by.sectors)

share.in.columns.extra

logs.matrix.kld<-matrix(0, ncol=m, nrow=n)

logs.matrix.kld<-shares.in.columns\*log(shares.in.columns/share.in.columns.extra)

logs.matrix.kld

sum.columns.kld<-colSums(logs.matrix.kld)

sum.columns.kld

equal.dist.H<-abs((1/n)\*log(1/n)\*n)

equal.dist.H

wynik[1,]<-abs(sum.columns)

wynik[2,]<-abs(sum.columns)/rep(equal.dist.H, times=m)

wynik[3,]<-(-abs(sum.columns))+rep(equal.dist.H, times=m)

wynik[4,]<-abs(sum.columns.kld)

wynik

wyniki.all[6:9,]<-wynik

################################\

**# diversification index**

wynik<-matrix(0, ncol=m, nrow=1)

colnames(wynik)<-paste("region",1:m)

rownames(wynik)<-c("index of diversification")

wynik

shares.in.columns<-matrix(0, ncol=m, nrow=n)

shares.in.columns

for(i in 1:n){

temp<-empl[i,]/sum.by.regions[1:m]

shares.in.columns[i,]<-as.matrix(temp[1,])}

shares.in.columns

colSums(shares.in.columns) # check of sums in columns

share.in.columns.extra<-sum.by.sectors/sum(sum.by.sectors)

share.in.columns.extra

hyp.con<-c(100, rep(0, times=n-1))

for(j in 1:m){

vec1<- shares.in.columns[,j]\*100

vec2<- share.in.columns.extra\*100

vec1

vec2

vec1.s<-sort(vec1, decreasing=TRUE)

vec2.s<-sort(vec2, decreasing=TRUE)

vec1.sc<-cumsum(vec1.s)

vec2.sc<-cumsum(vec2.s)

vec3.sc<-cumsum(hyp.con)

vec1.sum<-sum(vec1.sc)

vec2.sum<-sum(vec2.sc)

vec3.sum<-sum(vec3.sc)

vec1.sum

vec2.sum

vec3.sum

index<-(vec1.sum- vec2.sum)/( vec3.sum- vec2.sum)

index

wynik[1,j]<-index}

wynik

wyniki.all[10,]<-wynik

**################################\**

**# LQ, Hachman, RSI**

wynik<-matrix(0, ncol=m, nrow=2)

colnames(wynik)<-paste("region",1:m)

rownames(wynik)<-c("Hachman", "RSI (max LQ)")

wynik

shares.in.columns<-matrix(0, ncol=m, nrow=n)

shares.in.columns

for(i in 1:n){

temp<-empl[i,]/sum.by.regions[1:m]

shares.in.columns[i,]<-as.matrix(temp[1,])}

shares.in.columns

colSums(shares.in.columns) # check of sums in columns

share.in.columns.extra<-sum.by.sectors/sum(sum.by.sectors)

share.in.columns.extra

LQ<-matrix(0, ncol=m, nrow=n)

rownames(LQ)<-sectors

for(j in 1:m){

vec1<- shares.in.columns[,j]

vec2<- share.in.columns.extra

vec1

vec2

LQ[,j]<-vec1/vec2}

round(LQ,3)

temp<-LQ\*shares.in.columns

temp

temp1<-colSums(temp)

temp1

wynik[1,]<-1/temp1

wynik[2,]<-apply(LQ, 2, max) # max po kolumnach

wynik

wyniki.all[11:12,]<-wynik

**################################\**

**# Gini**

wynik<-matrix(0, ncol=m, nrow=1)

colnames(wynik)<-paste("region",1:m)

rownames(wynik)<-c("Gini")

wynik

shares.in.columns<-matrix(0, ncol=m, nrow=n)

shares.in.columns

for(i in 1:n){

temp<-empl[i,]/sum.by.regions[1:m]

shares.in.columns[i,]<-as.matrix(temp[1,])}

shares.in.columns

colSums(shares.in.columns) # check of sums in columns

share.in.columns.extra<-sum.by.sectors/sum(sum.by.sectors)

share.in.columns.extra

for(j in 1:m){

vec1<- shares.in.columns[,j]

vec2<- share.in.columns.extra

vec1

vec2

vec3<-vec1/vec2

vec3

vec3.av<-mean(vec3)

vec3.av

vec3.sort<-sort(vec3, decreasing=TRUE)

vec3.sort

vec4<-1:n

vec4

vec5<-abs(vec3.sort-vec3.av)\*vec4

tot.vec5<-sum(vec5)

tot.vec5

wynik[1,j]<-(tot.vec5\*2)/(n\*n\* vec3.av)}

wynik

wyniki.all[13,]<-wynik

**################################\**

**wyniki.all**

**LQ**

**################################\**

**####################################################################################\**

**# by regions for industries**

**##############################\**

**# defining the data - przykładowe**

setwd("E:/My all/&Naukowe/moje naukowe w toku/GRANT ERSA/# obliczenia cluster-based")

dane<-read.csv("example.csv", header=TRUE, dec=",", sep=";")

sum.by.regions<-dane[5, 2:8]

sum.by.sectors<-dane[1:4, 8]

sectors<-dane[1:4,1]

empl<-dane[1:4, 2:7]

m<-6 # numer of regions

n<-4 # numer of sectors

**##############################\**

**# defining the data on firms - przykładowe**

setwd("E:/My all/&Naukowe/moje naukowe w toku/GRANT ERSA/# obliczenia cluster-based")

firmy<-read.csv("firms.csv", header=TRUE, dec=",", sep=";")

dim(firmy)

firmy

sector size100 size50 size10 size1 Total

Industry1 0 1 6 9 119

Industry2 0 0 11 11 121

Industry3 0 0 9 15 105

Industry4 0 0 4 28 68

size<-matrix(0, nrow=dim(firmy)[1], ncol=dim(firmy)[2]-2)

size[,1]<-rep(100, times= dim(firmy)[1])

size[,2]<-rep(50, times= dim(firmy)[1])

size[,3]<-rep(10, times= dim(firmy)[1])

size[,4]<-rep(1, times= dim(firmy)[1])

size

**##############################\**

**# defining the distance between regions - przykładowe**

setwd("E:/My all/&Naukowe/moje naukowe w toku/GRANT ERSA/# obliczenia cluster-based")

dist<-read.csv("dist.csv", header=TRUE, dec=",", sep=";")

dim(dist)

colnames(dist)<-paste("region",1:m)

rownames(dist)<-paste("region",1:m)

dist

**##############################\**

**# wyniki zbiorczo**

wyniki.all<-matrix(0, ncol=n, nrow=14)

rownames(wyniki.all)<-c("Gini", "Krugman\_concentration\_index", "Bruelhart\_&\_Traeger", "locational\_Gini", "Agglomeration\_V", "Herfindahl", "Ellison-Glaeser", "Maurel-Sedillot", "Shannon’s\_H", "Relative\_H", "Theil’s\_H", "KLD", "Bergstrand\_clustering\_index","Moran’s\_I\_for\_LQ" )

wyniki.all

shares.in.rows<-matrix(0, ncol=m, nrow=n)

shares.in.rows

for(i in 1:m){

temp<-empl[,i]/sum.by.sectors[1:n]

shares.in.rows[,i]<-as.matrix(temp)}

shares.in.rows

rowSums(shares.in.rows) # check of sums in columns

share.in.rows.extra<-sum.by.regions/sum(sum.by.regions[1:m])

share.in.rows.extra

shares.in.columns<-matrix(0, ncol=m, nrow=n)

shares.in.columns

for(i in 1:n){

temp<-empl[i,]/sum.by.regions[1:m]

shares.in.columns[i,]<-as.matrix(temp[1,])}

shares.in.columns

colSums(shares.in.columns) # check of sums in columns

share.in.columns.extra<-sum.by.sectors/sum(sum.by.sectors)

share.in.columns.extra

**################################\**

**# Gini**

wynik<-matrix(0, ncol=n, nrow=1)

colnames(wynik)<-paste("sector",1:n)

rownames(wynik)<-c("Gini")

wynik

for(j in 1:n){

vec1<- shares.in.rows[j,]

vec2<- share.in.rows.extra[1:m]

vec1

vec2

vec3<-vec1/vec2

vec3

vec3.av<- rowMeans(vec3, na.rm=FALSE, dims=1)

vec3.av

vec3.sort<-sort(vec3, decreasing=TRUE)

vec3.sort

vec4<-1:m

vec4

vec5<-abs(vec3.sort-vec3.av)\*vec4

tot.vec5<-sum(vec5)

tot.vec5

wynik[1,j]<-(tot.vec5\*2)/(m\*m\* vec3.av)}

wynik

wyniki.all[1,]<-wynik

wyniki.all

**################################\**

**# Krugman concentration index**

wynik<-matrix(0, ncol=n, nrow=1)

colnames(wynik)<-paste("sector",1:n)

rownames(wynik)<-c("Krugman concentration index")

wynik

for(i in 1:n){

diff.abs<-abs(shares.in.rows[i,]-share.in.rows.extra[,1:m])

wynik[1,i]<-sum(diff.abs)}

wynik

wyniki.all[2,]<-wynik

wyniki.all

**################################\**

**# Bruelhart –Traeger index**

wynik<-matrix(0, ncol=n, nrow=1)

colnames(wynik)<-paste("sector",1:n)

rownames(wynik)<-c("Bruelhart & Traeger index")

wynik

temp<-matrix(0, nrow=n, ncol=m)

for(i in 1:n){

temp[i,]<-as.matrix(shares.in.rows[i,]/share.in.rows.extra[,1:m])}

temp2<-matrix(0, nrow=n, ncol=m)

for(i in 1:n){

a<- share.in.rows.extra\*temp[i,]\*log(temp[i,], base=10)

temp2[i,]<-as.matrix(a[1,1:m])}

temp2

wynik<-rowSums(temp2)

wynik

wyniki.all[3,]<-wynik

wyniki.all

**################################\**

**# locational Gini**

wynik<-matrix(0, ncol=n, nrow=1)

colnames(wynik)<-paste("sector",1:n)

rownames(wynik)<-c("locational Gini")

wynik

temp<-matrix(0, nrow=n, ncol=m)

for(i in 1:n){

temp[i,]<-as.matrix(shares.in.columns[i,]/share.in.rows.extra[,1:m])}

temp

for(i in 1:n){

temp2<-matrix(0, nrow=m+1, ncol=m+1)

temp2[1,2:(m+1)]<-temp[i,]

temp2[2:(m+1),1]<-temp[i,]

a5<-mean(temp[i,])

for(g in 1:m){

for(h in 1:m){

temp2[g+1, h+1]<-abs(temp2[1,h+1]-temp2[g+1,1])}}

temp2

temp3<-temp2[2:(m+1), 2:(m+1)]

temp3

a3<-sum(temp3)

a3

a4<-m\*(m-1)

a4

a5

G<-(a3/a4)/(n\*a5)

G

wynik[1,i]<-G

}

wynik

wyniki.all[4,]<-wynik

wyniki.all

**################################\**

**# Agglomeration V**

wynik<-matrix(0, ncol=n, nrow=1)

colnames(wynik)<-paste("sector",1:n)

rownames(wynik)<-c("Agglomeration V")

wynik

temp1<-rowSums(shares.in.columns)/m

temp1a<-sum(share.in.rows.extra[1:m])/m

temp1

temp1a

temp3<-matrix(0, ncol=m, nrow=n)

for(i in 1:m){

temp3[,i]<-(shares.in.columns[,i]-temp1)^2}

temp3<-as.data.frame(temp3)

temp3$total<-rowSums(temp3)

diff.share.row.sq<-(share.in.rows.extra[,1:m]-temp1a)^2

diff.share.row.sq

sum.diff.share.row.sq<-sum(diff.share.row.sq)

sum.diff.share.row.sq

temp4<-(sum.diff.share.row.sq/m)^0.5

temp4

temp3$temp<-(temp3$total/m)^0.5

temp3

temp3$Vi<-(temp3$temp/temp1)/(temp4/temp1a)

temp3

wynik[1,]<-temp3$Vi

wynik

wyniki.all[5,]<-wynik[1,]

wyniki.all

**################################\**

**# Herfindahl, Ellison-Glaeser**

wynik<-matrix(0, ncol=n, nrow=2)

colnames(wynik)<-paste("sector",1:n)

rownames(wynik)<-c("Herfinahl", "Ellison-Glaeser")

wynik

temp<-matrix(0, nrow=dim(firmy)[1], ncol=dim(firmy)[2]-2)

for(i in 1: dim(firmy)[1]-1){

temp[,i]<-(size[,i]/firmy[,7])^2\*firmy[,i+1]}

temp<-as.data.frame(temp)

temp$total<-rowSums(temp)

wynik[1,]<- temp$total

wynik

temp<-matrix(0, nrow=n, ncol=m)

for(i in 1:n){

temp[i,]<-as.matrix((shares.in.rows[i,]-share.in.rows.extra[,1:m])^2)}

temp<-as.data.frame(temp)

temp$total<-rowSums(temp)

sum.share.row.sq<-sum(share.in.rows.extra[,1:m]^2)

eg<-matrix(0, nrow=n, ncol=3)

for(i in 1:n){

eg[,1]<-temp$total-(1-sum.share.row.sq)\*wynik[1,]

eg[,2]<-(1- sum.share.row.sq)\*(1-wynik[1,])

eg[,3]<-eg[,1]/eg[,2]}

eg

wynik[2,]<-eg[,3]

wyniki.all[6:7,]<-wynik[1:2,]

wyniki.all

**################################\**

**# Maurel-Sedillot index**

wynik<-matrix(0, ncol=n, nrow=2)

colnames(wynik)<-paste("sector",1:n)

rownames(wynik)<-c("Maurel-Sedillot", "Herfindahl")

wynik

temp<-matrix(0, nrow=dim(firmy)[1], ncol=dim(firmy)[2]-2)

for(i in 1: dim(firmy)[1]-1){

temp[,i]<-(size[,i]/firmy[,7])^2\*firmy[,i+1]}

temp<-as.data.frame(temp)

temp$total<-rowSums(temp)

wynik[2,]<- temp$total

wynik

temp1<-shares.in.rows^2

temp1<-as.data.frame(temp1)

temp1$total<-rowSums(temp1)

temp1

temp1a<-share.in.rows.extra[1:m]^2

temp1a

sum.temp1a<-sum(temp1a)

sum.temp1a

ms<-matrix(0, nrow=n, ncol=3)

for(i in 1:n){

ms[,1]<-(temp1$total-sum.temp1a)/(1-sum.temp1a)-wynik[2,]

ms[,2]<-(1-wynik[2,])

ms[,3]<-ms[,1]/ms[,2]}

ms

wynik[1,]<-ms[,3]

wynik

wyniki.all[8,]<-wynik[1,]

wyniki.all

**################################\**

**# Entropy – Theil’s H, Shannon’s H, Relative H, KLD**

wynik<-matrix(0, ncol=n, nrow=4)

colnames(wynik)<-paste("sector",1:n)

rownames(wynik)<-c("Shannon’s H", "Relative H", "Theil’s H", "KLD")

wynik

logs.matrix<-matrix(0, ncol=m, nrow=n)

logs.matrix<-shares.in.rows\*log(shares.in.rows)

logs.matrix

sum.rows<-rowSums(logs.matrix)

sum.rows

exp.share<-rep(1/m, times=m)

exp.share

logs.matrix.kld<-matrix(0, ncol=m, nrow=n)

logs.matrix.kld<-shares.in.rows\*log(shares.in.rows/exp.share)

logs.matrix.kld

sum.rows.kld<-rowSums(logs.matrix.kld)

sum.rows.kld

equal.dist.H<-abs((1/m)\*log(1/m)\*m)

equal.dist.H

wynik[1,]<-abs(sum.rows)

wynik[2,]<-abs(sum.rows)/rep(equal.dist.H, times=n)

wynik[3,]<-(-abs(sum.rows))+rep(equal.dist.H, times=n)

wynik[4,]<-abs(sum.rows.kld)

wynik

wyniki.all[9:12,]<-wynik

wyniki.all

**################################\**

**# Clustering Index (Bergstrand)**

wynik<-matrix(0, ncol=n, nrow=1)

colnames(wynik)<-paste("sector",1:n)

rownames(wynik)<-c("Bergstrand clustering index")

wynik

dist<-dist+0.001

dist<-as.matrix(dist)

dist

for(i in 1:n){

# counter

temp2<-matrix(0, nrow=m+1, ncol=m+1)

temp2[1,2:(m+1)]<-shares.in.columns[i,]

temp2[2:(m+1),1]<-shares.in.columns[i,]

for(g in 1:m){

for(h in 1:m){

temp2[g+1, h+1]<-(temp2[1,h+1]\*temp2[g+1,1]/dist[g,h])}}

temp2

# nominator

temp3<-matrix(0, nrow=m+1, ncol=m+1)

temp3[1,2:(m+1)]<-as.matrix(share.in.rows.extra[1:m])

temp3[2:(m+1),1]<-as.matrix(share.in.rows.extra[1:m])

for(g in 1:m){

for(h in 1:m){

temp3[g+1, h+1]<-(temp3[1,h+1]\*temp3[g+1,1]/dist[g,h])}}

temp3

up<-sum(temp2[-1,-1])

down<-sum(temp3[-1,-1])

Bergs <-up/down

wynik[1,i]<-Bergs}

wynik

wyniki.all[13,]<-wynik

wyniki.all

**################################\**

**# Moran for LQ**

wynik<-matrix(0, ncol=n, nrow=1)

colnames(wynik)<-paste("sector",1:n)

rownames(wynik)<-c("Moran’s I for LQ")

wynik

library(spdep)

library(maptools)

library(sp)

library(rgdal)

map<-readOGR(".", "mapa6wybr") # 16 jedn.

projekcja<-"+proj=longlat +datum=WGS84"

map<-spTransform(map, CRS(projekcja))

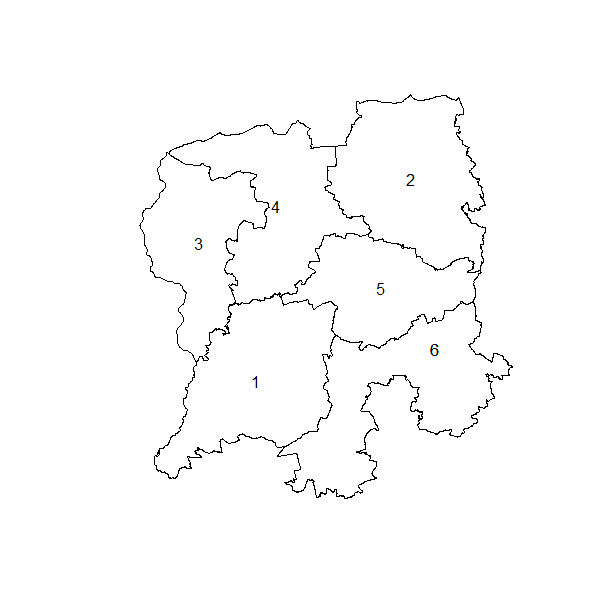
plot(map)

crdsb<-coordinates(map)

b<-1:6

text(crdsb[1:5,], labels=b[1:5])

text(15.7, 52.1, labels=b[6])



cont.nb<-poly2nb(as(map, "SpatialPolygons"))

cont.listw<-nb2listw(cont.nb, style="W")

cont.mat<-nb2mat(cont.nb)

cont.mat

[,1] [,2] [,3] [,4] [,5] [,6]

0 0.00 0.00 0.25 0.25 0.25 0.25

1 0.00 0.00 0.00 0.50 0.50 0.00

2 0.50 0.00 0.00 0.50 0.00 0.00

3 0.25 0.25 0.25 0.00 0.25 0.00

4 0.25 0.25 0.00 0.25 0.00 0.25

5 0.50 0.00 0.00 0.00 0.50 0.00

LQ<-matrix(0, ncol=m, nrow=n)

rownames(LQ)<-sectors

for(j in 1:m){

vec1<- shares.in.columns[,j]

vec2<- share.in.columns.extra

vec1

vec2

LQ[,j]<-vec1/vec2}

round(LQ,3)

for(i in 1:n){

mm<-moran.test(LQ[i,], cont.listw)

wynik[1,i]<-mm$estimate[1]}

wynik

wyniki.all[14,]<-wynik

wyniki.all

**################################\**

**# Arbia & Piras**

poss.rows<-40

wynik<-matrix(0, ncol=n, nrow=poss.rows)

colnames(wynik)<-paste("sector",1:n)

for(i in 1:n){

x<-as.matrix(dane[i,2:(m+1)])

a<-permutations(4, 4, v=x, set=FALSE, repeats.allowed=FALSE)

b<-matrix(0, nrow=dim(a)[1], ncol=2)

for(z in 1:dim(a)[1]){

mmm<-moran.test(a[z,], cont.listw)

b[z,1]<-as.numeric(mmm$estimate[1])

b[z,2]<-as.numeric(mmm$p.value)}

plot(density(b[,1]))

maks<-max(b[,1])

maks.list<-which(b[,1]==maks)

aver<-mean(a[1,])

down<-(dane[i,1:m+1]-rep(aver, times=m))^2

sum.down<-sum(down)

temp<-a[maks.list,]

temp2<-matrix(0, ncol=m, nrow=length(maks.list))

for(j in 1:length(maks.list)){

temp2[j,]<-as.matrix((temp[j,]-rep(aver, times=m))\*(dane[i,1:m+1]-rep(aver, times=m)))}

temp2<-as.data.frame(temp2)

temp2$sum<-rowSums(temp2)

temp2$lambda<-temp2$sum/sum.down

wynik[,i]<-c(temp2$lambda, rep(NA, times=poss.rows-dim(temp2)[1]))}

wynik