Supplementary material to paper

"Cluster-based measures of regional concentration. Critical overview"

Katarzyna Kopczewska
University of Warsaw, Faculty of Economic Sciences, Ul. Dluga 44/50, 00-241 Warszawa, Poland
e-mail: kkopczewska@wne.uw.edu.pl
ORCID: http://orcid.org/0000-0003-1065-1790

The supplementary file contains the R Codes to run the simulation presented in the paper. It includes the codes for the measures of geographical and sectoral concentration. It allows for replicating all the figures and computations presented in the paper.

Content

```
# generating random data & matrices – multi matrix operation in loop

## set seed for reproducibility
set.seed(1)

# size of generated matrix – big table
m<-25 # number of regions - number of columns (vertical dim)
n<-100 # number of sectors - number of rows (horizontal dim)

# generating matrix: double randomization – only for big table
for(t in 1:1000){
vec1<-rnorm(m*n, runif(m*n, min=0, max=50), runif(m*n, min=0, max=25))
vec1[vec1<2]<-2 # no negative values, truncated in 1 (for logs)
m1<-matrix(vec1, nrow=n, ncol=m, byrow=TRUE)
assign(paste("m1.dr.b.", t, sep=""), m1)} # m-matrix, dr-double randomisation, b-big

# generating matrix: single randomization  – only for big table
mi2<-runif(m*n, min=0, max=50)
sigma2<-runif(m*n, min=0, max=25)
# loop from here, mi & sigma drawn once only
for(t in 1:1000){
vec2<-rnorm(m*n, mi2, sigma2)
vec2[vec2<2]<-2 # no negative values, truncated in 0
m2<-matrix(vec2, nrow=n, ncol=m, byrow=TRUE)
assign(paste("m1.sr.b.", t, sep=""), m2)} # m-matrix, sr-single randomisation, b-big

# generating matrix: permutation  – only for big table
mi3<-runif(m*n, min=0, max=50)
sigma3<-runif(m*n, min=0, max=25)
norm3<-rnorm(m*n, mi3, sigma3)
norm3[norm3<2]<-2 # no negative values, truncated in 1 (for logs in entropy)
# loop from here, mi & sigma & norm.dist drawn once only
for(t in 1:1000){
vec3<-sample(norm3, n*m, replace = FALSE)
m3<-matrix(vec3, nrow=n, ncol=m, byrow=TRUE)
assign(paste("m1.p.b.", t, sep=""), m3)} # m-matrix, p-permutation, b-big

# aggregation of big table to middle table
# middle table
#m<-10 # number of regions - number of columns (vertical dim)
#n<-25 # number of sectors - number of rows (horizontal dim)
vec.agr.m<-c(1,1,1,1,1,2,2,2,2,3,3,3,4,4,5,6,6,7,7,7,8,8,8,9,10) # irregular clusters
vec.agr.n1<-c(1,1,1,1,1,2,2,2,2,3,3,3,4,4,5,6,6,7,7,7,8,8,8,9,10)+rep(c(0,10), each=25)
# irregular clusters
vec.agr.n2<-rep(21:25, each=10)
vec.agr.n<-c(vec.agr.n1, vec.agr.n2) # irregular clusters

for(t in 1:1000){
m.agr1<-aggregate(get(paste("m1.dr.b.",t,sep="")), by=list(vec.agr.n), sum)
m.agr2<-aggregate(t(m.agr1)[2:26,], by=list(vec.agr.m), sum)
m.m<-t(m.agr2)[2:26,]
assign(paste("m1.dr.m.", t, sep=""), m.m)} # m-matrix, dr-double randomisation, m-middle

# aggregation of big table to middle table
# middle table
#m<-10 # number of regions - number of columns (vertical dim)
#n<-25 # number of sectors - number of rows (horizontal dim)
vec.agr.m<-c(1,1,1,1,1,2,2,2,2,3,3,3,4,4,5,6,6,7,7,7,8,8,8,9,10) # irregular clusters
vec.agr.n1<-c(1,1,1,1,1,2,2,2,2,3,3,3,4,4,5,6,6,7,7,7,8,8,8,9,10)+rep(c(0,10), each=25)
# irregular clusters
vec.agr.n2<-rep(21:25, each=10)
vec.agr.n<-c(vec.agr.n1, vec.agr.n2) # irregular clusters

for(t in 1:1000){
```

```
m.agr1<-aggregate(get(paste("m1.sr.b.",t,sep="")), by=list(vec.agr.n), sum)
m.agr2<-aggregate(t(m.agr1)[2:26,], by=list(vec.agr.m), sum)
m.m<-t(m.agr2)[2:26,]
assign(paste("m1.sr.m.", t, sep=""), m.m)} # m-matrix, sr-single randomisation, m-middle

# aggregation of big table to middle table
# middle table
#m<-10 # number of regions - number of columns (vertical dim)
#n<-25 # number of sectors - number of rows (horizontal dim)
vec.agr.m<-c(1,1,1,1,1,2,2,2,2,3,3,3,4,4,5,6,6,7,7,7,8,8,8,9,10) # irregular clusters
vec.agr.n1<-c(1,1,1,1,1,2,2,2,2,3,3,3,4,4,5,6,6,7,7,7,8,8,8,9,10)+rep(c(0,10), each=25)
# irregular clusters
vec.agr.n2<-rep(21:25, each=10)
vec.agr.n<-c(vec.agr.n1, vec.agr.n2) # irregular clusters

for(t in 1:1000){
m.agr1<-aggregate(get(paste("m1.p.b.",t,sep="")), by=list(vec.agr.n), sum)
m.agr2<-aggregate(t(m.agr1)[2:26,], by=list(vec.agr.m), sum)
m.m<-t(m.agr2)[2:26,]
assign(paste("m1.p.m.", t, sep=""), m.m)} # m-matrix, p-permutation, m-middle

# aggregation of middle table to small table
# size of generated matrix
# small table
#m<-5 # number of regions - number of columns (vertical dim)
#n<-5 # number of sectors - number of rows (horizontal dim)
vec.agr.m<-rep(1:5, each=2)
vec.agr.n<-rep(1:5, each=5)
for(t in 1:1000){
m.agr1<-aggregate(get(paste("m1.dr.m.",t, sep="")), by=list(vec.agr.n), sum)
m.agr2<-aggregate(t(m.agr1)[2:11,], by=list(vec.agr.m), sum)
m.s<-t(m.agr2)[2:6,]
assign(paste("m1.dr.s.", t, sep=""), m.s)} # m-matrix, dr-double randomization, s-small

# aggregation of middle table to small table
# size of generated matrix
# small table
#m<-5 # number of regions - number of columns (vertical dim)
#n<-5 # number of sectors - number of rows (horizontal dim)
vec.agr.m<-rep(1:5, each=2)
vec.agr.n<-rep(1:5, each=5)
for(t in 1:1000){
m.agr1<-aggregate(get(paste("m1.sr.m.",t,sep="")), by=list(vec.agr.n), sum)
m.agr2<-aggregate(t(m.agr1)[2:11,], by=list(vec.agr.m), sum)
m.s<-t(m.agr2)[2:6,]
assign(paste("m1.sr.s.", t, sep=""), m.s)} # m-matrix, sr-double randomization, s-small

# aggregation of middle table to small table
# size of generated matrix
# small table
#m<-5 # number of regions - number of columns (vertical dim)
#n<-5 # number of sectors - number of rows (horizontal dim)
vec.agr.m<-rep(1:5, each=2)
vec.agr.n<-rep(1:5, each=5)
for(t in 1:1000){
m.agr1<-aggregate(get(paste("m1.p.m.",t,sep="")), by=list(vec.agr.n), sum)
m.agr2<-aggregate(t(m.agr1)[2:11,], by=list(vec.agr.m), sum)
m.s<-t(m.agr2)[2:6,]
assign(paste("m1.p.s.", t, sep=""), m.s)} # m-matrix, p-permutation, s-small

# once again, but for different distribution parameters - M2
# size of generated matrix - big table
m<-25 # number of regions - number of columns (vertical dim)
n<-100 # number of sectors - number of rows (horizontal dim)
```

```r
# generating matrix: double randomization – only for big table
for(t in 1:1000){
vec1<-rnorm(m*n, runif(m*n, min=0, max=500), runif(m*n, min=0, max=100))
vec1[vec1<2]<-2 # no negative values, truncated in 0
m1<-matrix(vec1, nrow=n, ncol=m, byrow=TRUE)
assign(paste("m2.dr.b.", t, sep=""), m1)} # m-matrix, dr-double randomisation, b-big

# generating matrix: single randomization  – only for big table
mi2<-runif(m*n, min=0, max=500)
sigma2<-runif(m*n, min=0, max=100)
# loop from here, mi & sigma drawn once only
for(t in 1:1000){
vec2<-rnorm(m*n, mi2, sigma2)
vec2[vec2<2]<-2 # no negative values, truncated in 0
m2<-matrix(vec2, nrow=n, ncol=m, byrow=TRUE)
assign(paste("m2.sr.b.", t, sep=""), m2)} # m-matrix, sr-single randomisation, b-big

# generating matrix: permutation  – only for big table
mi3<-runif(m*n, min=0, max=500)
sigma3<-runif(m*n, min=0, max=100)
norm3<-rnorm(m*n, mi3, sigma3)
norm3[norm3<2]<-2 # no negative values, truncated in 1 (for logs in entropy)
# loop from here, mi & sigma & norm.dist drawn once only
for(t in 1:1000){
vec3<-sample(norm3, n*m, replace = FALSE)
m3<-matrix(vec3, nrow=n, ncol=m, byrow=TRUE)
assign(paste("m2.p.b.", t, sep=""), m3)} # m-matrix, p-permutation, b-big

# aggregation of big table to middle table
# middle table
#m<-10 # number of regions - number of columns (vertical dim)
#n<-25 # number of sectors - number of rows (horizontal dim)
vec.agr.m<-c(1,1,1,1,1,2,2,2,2,3,3,3,4,4,5,6,6,7,7,7,8,8,8,9,10) # irregular clusters
vec.agr.n1<-c(1,1,1,1,1,2,2,2,2,3,3,3,4,4,5,6,6,7,7,7,8,8,8,9,10)+rep(c(0,10), each=25)
# irregular clusters
vec.agr.n2<-rep(21:25, each=10)
vec.agr.n<-c(vec.agr.n1, vec.agr.n2) # irregular clusters

for(t in 1:1000){
m.agr1<-aggregate(get(paste("m2.dr.b.",t,sep="")), by=list(vec.agr.n), sum)
m.agr2<-aggregate(t(m.agr1)[2:26,], by=list(vec.agr.m), sum)
m.m<-t(m.agr2)[2:26,]
assign(paste("m2.dr.m.", t, sep=""), m.m)} # m-matrix, dr-double randomisation, m-
middle

# aggregation of big table to middle table
# middle table
#m<-10 # number of regions - number of columns (vertical dim)
#n<-25 # number of sectors - number of rows (horizontal dim)
vec.agr.m<-c(1,1,1,1,1,2,2,2,2,3,3,3,4,4,5,6,6,7,7,7,8,8,8,9,10) # irregular clusters
vec.agr.n1<-c(1,1,1,1,1,2,2,2,2,3,3,3,4,4,5,6,6,7,7,7,8,8,8,9,10)+rep(c(0,10), each=25)
# irregular clusters
vec.agr.n2<-rep(21:25, each=10)
vec.agr.n<-c(vec.agr.n1, vec.agr.n2) # irregular clusters
for(t in 1:1000){
m.agr1<-aggregate(get(paste("m2.sr.b.",t,sep="")), by=list(vec.agr.n), sum)
m.agr2<-aggregate(t(m.agr1)[2:26,], by=list(vec.agr.m), sum)
m.m<-t(m.agr2)[2:26,]
assign(paste("m2.sr.m.", t, sep=""), m.m)} # m-matrix, sr-single randomisation, m-
middle

# aggregation of big table to middle table
# middle table
#m<-10 # number of regions - number of columns (vertical dim)
```

```
#n<-25 # number of sectors - number of rows (horizontal dim)
vec.agr.m<-c(1,1,1,1,1,2,2,2,2,3,3,3,4,4,5,6,6,7,7,7,8,8,8,9,10) # irregular clusters
vec.agr.n1<-c(1,1,1,1,1,2,2,2,2,3,3,3,4,4,5,6,6,7,7,7,8,8,8,9,10)+rep(c(0,10), each=25)
# irregular clusters
vec.agr.n2<-rep(21:25, each=10)
vec.agr.n<-c(vec.agr.n1, vec.agr.n2) # irregular clusters
for(t in 1:1000){
m.agr1<-aggregate(get(paste("m2.p.b.",t,sep="")), by=list(vec.agr.n), sum)
m.agr2<-aggregate(t(m.agr1)[2:26,], by=list(vec.agr.m), sum)
m.m<-t(m.agr2)[2:26,]
assign(paste("m2.p.m.", t, sep=""), m.m)} # m-matrix, p-permutation, m-middle

# aggregation of middle table to small table
# size of generated matrix
# small table
#m<-5 # number of regions - number of columns (vertical dim)
#n<-5 # number of sectors - number of rows (horizontal dim)
vec.agr.m<-rep(1:5, each=2)
vec.agr.n<-rep(1:5, each=5)
for(t in 1:1000){
m.agr1<-aggregate(get(paste("m2.dr.m.",t, sep="")), by=list(vec.agr.n), sum)
m.agr2<-aggregate(t(m.agr1)[2:11,], by=list(vec.agr.m), sum)
m.s<-t(m.agr2)[2:6,]
assign(paste("m2.dr.s.", t, sep=""), m.s)} # m-matrix, dr-double randomization, s-small

# aggregation of middle table to small table
# size of generated matrix
# small table
#m<-5 # number of regions - number of columns (vertical dim)
#n<-5 # number of sectors - number of rows (horizontal dim)
vec.agr.m<-rep(1:5, each=2)
vec.agr.n<-rep(1:5, each=5)
for(t in 1:1000){
m.agr1<-aggregate(get(paste("m2.sr.m.",t,sep="")), by=list(vec.agr.n), sum)
m.agr2<-aggregate(t(m.agr1)[2:11,], by=list(vec.agr.m), sum)
m.s<-t(m.agr2)[2:6,]
assign(paste("m2.sr.s.", t, sep=""), m.s)} # m-matrix, sr-double randomization, s-small

# aggregation of middle table to small table
# size of generated matrix
# small table
#m<-5 # number of regions - number of columns (vertical dim)
#n<-5 # number of sectors - number of rows (horizontal dim)
vec.agr.m<-rep(1:5, each=2)
vec.agr.n<-rep(1:5, each=5)
for(t in 1:1000){
m.agr1<-aggregate(get(paste("m2.p.m.",t,sep="")), by=list(vec.agr.n), sum)
m.agr2<-aggregate(t(m.agr1)[2:11,], by=list(vec.agr.m), sum)
m.s<-t(m.agr2)[2:6,]
assign(paste("m2.p.s.", t, sep=""), m.s)} # m-matrix, p-permutation, s-small

# the code above in this section generated 18000 matrices of input data, which will be
used further
# their name follows the structure: m. set of parameters. type of drawing . size .
iteration
# so m1.dr.b.999 is for big matrix with double randomization iteration 999, mi(0,50),
sigma(0,25)
# and m2.dr.b.999 is for big matrix with double randomization iteration 999, mi(0,500),
sigma(0,100)
```

Distribution	Type of sampling	“Big” table *m=25* regions, *n=100* sectors	“Middle” table *m=10* regions, *n=25* sectors	“Small” table *m=5* regions, *n=5* sectors
		drawn t=1000 iterations	aggregation of the “big” table,	aggregation of “middle” table,

		(separate data tables) of 25·100=2500 cells	irregular groups of regions and sectors	irregular groups of regions and sectors
N_1(μ∈(0,50), σ∈(0,25))	Double randomisation	Scheme 1 → SC1, GC1 From: m1.dr.b.1 To: m1.dr.b.1000	Scheme 7 → SC7, GC7 From: m1.dr.m.1 To: m1.dr.m.1000	Scheme 13 → SC13, GC13 From: m1.dr.s.1 To: m1.dr.s.1000
N_1(μ∈(0,50), σ∈(0,25))	Single randomisation	Scheme 2 → SC2, GC2 From: m1.sr.b.1 To: m1.sr.b.1000	Scheme 8 → SC8, GC8 From: m1.sr.m.1 To: m1.sr.m.1000	Scheme 14 → SC14, GC14 From: m1.sr.s.1 To: m1.sr.s.1000
N_1(μ∈(0,50), σ∈(0,25))	Permutation	Scheme 3 → SC3, GC3 From: m1.p.b.1 To: m1.p.b.1000	Scheme 9 → SC9, GC9 From: m1.p.m.1 To: m1.p.m.1000	Scheme 15 → SC15, GC15 From: m1.p.s.1 To: m1.p.s.1000
N_2(μ∈(0,500), σ∈(0,100))	Double randomisation	Scheme 4 → SC4, GC4 From: m2.dr.b.1 To: m2.dr.b.1000	Scheme 10 → SC10, GC10 From: m2.dr.m.1 To: m2.dr.m.1000	Scheme 16 → SC16, GC16 From: m2.dr.s.1 To: m2.dr.s.1000
N_2(μ∈(0,500), σ∈(0,100))	Single randomisation	Scheme 5 → SC5, GC5 From: m2.sr.b.1 To: m2.sr.b.1000	Scheme 11 → SC11, GC11 From: m2.sr.m.1 To: m2.sr.m.1000	Scheme 17 → SC17, GC17 From: m2.sr.s.1 To: m2.sr.s.1000
N_2(μ∈(0,500), σ∈(0,100))	Permutation	Scheme 6 → SC6, GC6 From: m2.p.b.1 To: m2.p.b.1000	Scheme 12 → SC12, GC12 From: m2.p.m.1 To: m2.p.m.1000	Scheme 18 → SC18, GC18 From: m2.p.s.1 To: m2.p.s.1000

```
# calculating SC sectoral concentration measures

# for each matrix generated above and collecting the measures in matrices
# generating empty matrices for measures for each scheme (in total 18 matrices of SC measures)

# SC1 is for scheme 1

# for big tables – m=25 regions
m=25
t=1000
for(i in 1:6){
wyniki.all<-matrix(0, ncol=13*m, nrow=t)
a<-paste(rep(c("Ogive", "Krugman dissimilarity", "National Averages (NAI)", "Hallet", "Relative Diversity (RDI)", "Shannon’s H", "Relative H", "Theil’s H", "Kullback-Leibler divergence (KLD)", "Refined diversification", "Hachman", "Relative Specialisation (RSI, max LQ)", "Gini"), each=m),rep(1:m, times=13))
colnames(wyniki.all)<-a
assign(paste("SC", i, sep=""), wyniki.all)}

# for middle tables – m=10 regions
m=10
t=1000
for(i in 7:12){
wyniki.all<-matrix(0, ncol=13*m, nrow=t)
a<-paste(rep(c("Ogive", "Krugman dissimilarity", "National Averages (NAI)", "Hallet", "Relative Diversity (RDI)", "Shannon’s H", "Relative H", "Theil’s H", "Kullback-Leibler divergence (KLD)", "Refined diversification", "Hachman", "Relative Specialisation (RSI, max LQ)", "Gini"), each=m),rep(1:m, times=13))
colnames(wyniki.all)<-a
assign(paste("SC", i, sep=""), wyniki.all)}

# for small tables – m=5 regions
m=5
t=1000
for(i in 13:18){
wyniki.all<-matrix(0, ncol=13*m, nrow=t)
a<-paste(rep(c("Ogive", "Krugman dissimilarity", "National Averages (NAI)", "Hallet", "Relative Diversity (RDI)", "Shannon’s H", "Relative H", "Theil’s H", "Kullback-Leibler divergence (KLD)", "Refined diversification", "Hachman", "Relative Specialisation (RSI, max LQ)", "Gini"), each=m),rep(1:m, times=13))
colnames(wyniki.all)<-a
assign(paste("SC", i, sep=""), wyniki.all)}



# “FUNCTION” for SC measures
# operates on object „data”,
# requires setting number of rows of matrix (n=number of sectors)
# and number of columns of matrix (m=number of regions)
# and id of iteration (t from 1 to ex.1000)
# and the name of final object fin_obj
# pastes the results to prepared SC matrices, for each of t iterations


SCmeasures<-function(dane, n, m, t, fin_obj) {

#dane<-matrix(data, nrow=n, ncol=m,  byrow = FALSE)
rownames(dane)<-paste(rep("sector", times=n),1:n)
colnames(dane)<-paste(rep("region", times=m), 1:m)
sum.by.regions<-colSums(dane)
dane<-rbind(dane, sum.by.regions)
sum.by.sectors<-rowSums(dane)
dane<-cbind(dane, sum.by.sectors)
dane
```

```
empl<-dane[1:n, 1:m]
empl
sectors<-rownames(dane[1:n,])
sectors

# data ranges for output
range.Ogive<-1:m
range.Krugman<-(m+1):(2*m)
range.NAI<-(2*m+1):(3*m)
range.Hallet<-(3*m+1):(4*m)
range.RDI<-(4*m+1):(5*m)
range.Shannon<-(5*m+1):(6*m)
range.RelativeH<-(6*m+1):(7*m)
range.Theil<-(7*m+1):(8*m)
range.KLD<-(8*m+1):(9*m)
range.divers<-(9*m+1):(10*m)
range.Hachman<-(10*m+1):(11*m)
range.RSI<-(11*m+1):(12*m)
range.Gini<-(12*m+1):(13*m)

################################\
# common elements of calculations
# matrix of industrial shares by regions
shares.in.columns<-matrix(0, ncol=m, nrow=n)
for(i in 1:n){
temp<-empl[i,]/sum.by.regions[1:m]
shares.in.columns[i,]<-as.matrix(temp)}
shares.in.columns
colSums(shares.in.columns) # check of sums in columns

# vector of industrial shares for national economy
share.in.columns.extra<-sum.by.sectors/sum.by.sectors[n+1]
share.in.columns.extra

################################\
# Ogive index
wynik<-matrix(0, ncol=m, nrow=1)
colnames(wynik)<-paste("region",1:m)
rownames(wynik)<-c("Ogive index")

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
fin_obj[t,range.Ogive]<-wynik

################################\
# Krugman dissimilarity index, NAI, Hallet, Relative Diversity Index
wynik<-matrix(0, ncol=m, nrow=4)
colnames(wynik)<-paste("region",1:m)
rownames(wynik)<-c("Krugman Dissimilarity", "National Averages (NAI)", "Hallet",
"Relative Diversity (RDI)")
for(i in 1:m){
diff.abs<-abs(shares.in.columns[,i]- share.in.columns.extra[1:n])
wynik[1,i]<-sum(diff.abs)}

for(i in 1:m){
diff.abs<-(shares.in.columns[,i]- share.in.columns.extra[1:n])^2/
share.in.columns.extra[1:n]
wynik[2,i]<-sum(diff.abs)}
wynik[3,]<-wynik[1,]/2
```

```
wynik[4,]<-1/wynik[1,]
wynik
fin_obj[t,range.Krugman]<-wynik[1,]
fin_obj[t,range.NAI]<-wynik[2,]
fin_obj[t,range.Hallet]<-wynik[3,]
fin_obj[t,range.RDI]<-wynik[4,]

#################################\
# 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")

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

logs.matrix.kld<-matrix(0, ncol=m, nrow=n)
logs.matrix.kld<-shares.in.columns* log(shares.in.columns/ share.in.columns.extra)
sum.columns.kld<-colSums(logs.matrix.kld)
equal.dist.H<-abs((1/n)*log(1/n)*n)

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
fin_obj[t,range.Shannon]<-wynik[1,]
fin_obj[t,range.RelativeH]<-wynik[2,]
fin_obj[t,range.Theil]<-wynik[3,]
fin_obj[t,range.KLD]<-wynik[4,]

#################################\
# diversification index
wynik<-matrix(0, ncol=m, nrow=1)
colnames(wynik)<-paste("region",1:m)
rownames(wynik)<-c("Refined Diversification")

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.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)
index<-(vec1.sum- vec2.sum)/( vec3.sum- vec2.sum)
wynik[1,j]<-index}
wynik
fin_obj[t,range.divers]<-wynik

#################################\
# LQ, Hachman, RSI
wynik<-matrix(0, ncol=m, nrow=2)
colnames(wynik)<-paste("region",1:m)
rownames(wynik)<-c("Hachman", "Relative Specialisation (RSI, max LQ)")

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[1:n]
LQ[,j]<-vec1/vec2}

temp<-LQ*shares.in.columns
temp1<-colSums(temp)
wynik[1,]<-1/temp1
wynik[2,]<-apply(LQ, 2, max) # max po kolumnach
wynik
fin_obj[t,range.Hachman]<-wynik[1,]
fin_obj[t,range.RSI]<-wynik[2,]

################################\
# Gini
wynik<-matrix(0, ncol=m, nrow=1)
colnames(wynik)<-paste("region",1:m)
rownames(wynik)<-c("Gini")

for(j in 1:m){
vec1<- shares.in.columns[,j]
vec2<- share.in.columns.extra
vec3<-vec1/vec2
vec3.av<-mean(vec3)
vec3.sort<-sort(vec3, decreasing=TRUE)
vec4<-1:n
vec5<-abs(vec3.sort-vec3.av)*vec4
tot.vec5<-sum(vec5)
wynik[1,j]<-(tot.vec5*2)/(n*n* vec3.av)}
wynik
fin_obj[t,range.Gini]<-wynik
return(fin_obj)[t,]

} # closing the loop for the function
################################\



# calling SCmeasures() function  “(SCmeasures<-function(dane, n, m, t, fin_obj))”

# big tables
fin_obj<-SC1
for(t in 1:1000){
SC1[t,]<-SCmeasures(get(paste("m1.dr.b.",t, sep="")), 100, 25, t, fin_obj=SC1)[t,]}
summary(SC1)

fin_obj<-SC2
for(t in 1:1000){
SC2[t,]<-SCmeasures(get(paste("m1.sr.b.",t, sep="")), 100, 25, t, fin_obj=SC2)[t,]}
summary(SC2)

fin_obj<-SC3
for(t in 1:1000){
SC3[t,]<-SCmeasures(get(paste("m1.p.b.",t, sep="")), 100, 25, t, fin_obj=SC3)[t,]}
summary(SC3)

fin_obj<-SC4
for(t in 1:1000){
SC4[t,]<-SCmeasures(get(paste("m2.dr.b.",t, sep="")), 100, 25, t, fin_obj=SC4)[t,]}
summary(SC4)

fin_obj<-SC5
for(t in 1:1000){
```

```
SC5[t,]<-SCmeasures(get(paste("m2.sr.b.",t, sep="")), 100, 25, t, fin_obj=SC5)[t,]}
summary(SC5)

fin_obj<-SC6
for(t in 1:1000){
SC6[t,]<-SCmeasures(get(paste("m2.p.b.",t, sep="")), 100, 25, t, fin_obj=SC6)[t,]}
summary(SC6)

# middle tables
fin_obj<-SC7
for(t in 1:1000){
SC7[t,]<-SCmeasures(get(paste("m1.dr.m.",t, sep="")), 25, 10, t, fin_obj=SC7)[t,]}
summary(SC7)

fin_obj<-SC8
for(t in 1:1000){
SC8[t,]<-SCmeasures(get(paste("m1.sr.m.",t, sep="")), 25, 10, t, fin_obj=SC8)[t,]}
summary(SC8)

fin_obj<-SC9
for(t in 1:1000){
SC9[t,]<-SCmeasures(get(paste("m1.p.m.",t, sep="")), 25, 10, t, fin_obj=SC9)[t,]}
summary(SC9)

fin_obj<-SC10
for(t in 1:1000){
SC10[t,]<-SCmeasures(get(paste("m2.dr.m.",t, sep="")), 25, 10, t, fin_obj=SC10)[t,]}
summary(SC10)

fin_obj<-SC11
for(t in 1:1000){
SC11[t,]<-SCmeasures(get(paste("m2.sr.m.",t, sep="")), 25, 10, t, fin_obj=SC11)[t,]}
summary(SC11)

fin_obj<-SC12
for(t in 1:1000){
SC12[t,]<-SCmeasures(get(paste("m2.p.m.",t, sep="")), 25, 10, t, fin_obj=SC12)[t,]}
summary(SC12)

# small tables
fin_obj<-SC13
for(t in 1:1000){
SC13[t,]<-SCmeasures(get(paste("m1.dr.s.",t, sep="")), 5, 5, t, fin_obj=SC13)[t,]}
summary(SC13)

fin_obj<-SC14
for(t in 1:1000){
SC14[t,]<-SCmeasures(get(paste("m1.sr.s.",t, sep="")), 5, 5, t, fin_obj=SC14)[t,]}
summary(SC14)

fin_obj<-SC15
for(t in 1:1000){
SC15[t,]<-SCmeasures(get(paste("m1.p.s.",t, sep="")), 5, 5, t, fin_obj=SC15)[t,]}
summary(SC15)

fin_obj<-SC16
for(t in 1:1000){
SC16[t,]<-SCmeasures(get(paste("m2.dr.s.",t, sep="")), 5, 5, t, fin_obj=SC16)[t,]}
summary(SC16)

fin_obj<-SC17
for(t in 1:1000){
SC17[t,]<-SCmeasures(get(paste("m2.sr.s.",t, sep="")), 5, 5, t, fin_obj=SC17)[t,]}
summary(SC17)
```

```
fin_obj<-SC18
for(t in 1:1000){
SC18[t,]<-SCmeasures(get(paste("m2.p.s.",t, sep="")), 5, 5, t, fin_obj=SC18)[t,]}
summary(SC18)


# stack of SC measures matrx – one SC measure in one column

size<-rep(c(25,10,5), each=6)

for(i in 1:18){
temp<-get(paste("SC", i, sep=""))
m<-size[i]

range.Ogive<-1:m
s1<-temp[,range.Ogive]
ss1<-stack(as.data.frame(s1))
range.Krugman<-(m+1):(2*m)
s2<-temp[,range.Krugman]
ss2<-stack(as.data.frame(s2))
range.NAI<-(2*m+1):(3*m)
s3<-temp[,range.NAI]
ss3<-stack(as.data.frame(s3))
range.Hallet<-(3*m+1):(4*m)
s4<-temp[,range.Hallet]
ss4<-stack(as.data.frame(s4))
range.RDI<-(4*m+1):(5*m)
s5<-temp[,range.RDI]
ss5<-stack(as.data.frame(s5))
range.Shannon<-(5*m+1):(6*m)
s6<-temp[,range.Shannon]
ss6<-stack(as.data.frame(s6))
range.RelativeH<-(6*m+1):(7*m)
s7<-temp[,range.RelativeH]
ss7<-stack(as.data.frame(s7))
range.Theil<-(7*m+1):(8*m)
s8<-temp[,range.Theil]
ss8<-stack(as.data.frame(s8))
range.KLD<-(8*m+1):(9*m)
s9<-temp[,range.KLD]
ss9<-stack(as.data.frame(s9))
range.divers<-(9*m+1):(10*m)
s10<-temp[,range.divers]
ss10<-stack(as.data.frame(s10))
range.Hachman<-(10*m+1):(11*m)
s11<-temp[,range.Hachman]
ss11<-stack(as.data.frame(s11))
range.RSI<-(11*m+1):(12*m)
s12<-temp[,range.RSI]
ss12<-stack(as.data.frame(s12))
range.Gini<-(12*m+1):(13*m)
s13<-temp[,range.Gini]
ss13<-stack(as.data.frame(s13))

temp.stack<-cbind(ss1$values, ss2$values, ss3$values, ss4$values, ss5$values,
ss6$values, ss7$values, ss8$values, ss9$values, ss10$values, ss11$values, ss12$values,
ss13$values)
colnames(temp.stack)<-c("Ogive", "Krugman Dissimilarity", "National Averages (NAI)",
"Hallet", "Relative Diversification (RDI)", "Shannon’s H", "Relative H", "Theil’s H",
"KLD", "Refined Diversification", "Hachman", "Relative Specialisation (RSI, max LQ)",
"Gini")

assign(paste("SC", i, ".stack", sep=""), temp.stack)}

# call it e.g. SC1.stack
```

```
# calculating GC sectoral concentration measures

# for each matrix generated above and collecting the measures in matrices
# generating empty matrices for measures for each scheme (in total 18 matrices of GC
measures)

# GC1 is for scheme 1

# for big tables - n=100 industries
n=100
t=1000
for(i in 1:6){
wyniki.all<-matrix(0, ncol=9*n, nrow=t)
a<-paste(rep(c("Gini", "Krugman Concentration", "Bruelhart-Traeger", "locational Gini",
"Agglomeration V", "Shannon’s H", "Relative H", "Theil’s H", "KLD"), each=n),rep(1:n,
times=9))
colnames(wyniki.all)<-a
assign(paste("GC", i, sep=""), wyniki.all)}

# for middle tables - n=25 industries
n=25
t=1000
for(i in 7:12){
wyniki.all<-matrix(0, ncol=9*n, nrow=t)
a<-paste(rep(c("Gini", "Krugman Concentration", "Bruelhart-Traeger", "locational Gini",
"Agglomeration V", "Shannon’s H", "Relative H", "Theil’s H", "KLD"), each=n),rep(1:n,
times=9))
colnames(wyniki.all)<-a
assign(paste("GC", i, sep=""), wyniki.all)}

# for small tables - n=5 industries
n=5
t=1000
for(i in 13:18){
wyniki.all<-matrix(0, ncol=9*n, nrow=t)
a<-paste(rep(c("Gini", "Krugman Concentration", "Bruelhart-Traeger", "locational Gini",
"Agglomeration V", "Shannon’s H", "Relative H", "Theil’s H", "KLD"), each=n),rep(1:n,
times=9))
colnames(wyniki.all)<-a
assign(paste("GC", i, sep=""), wyniki.all)}

# “FUNCTION” for GC measures

# operates on object „data”,
# requires setting number of rows of matrix (n=number of sectors)
# and number of rows of columns (m=number of regions)
# and id of iteration (t from 1 to ex.1000)
# and the name of final object fin_obj
# pastes the results to prepared GC matrices, for each of t iterations

GCmeasures<-function(dane, n, m, t, fin_obj) {

#dane<-matrix(data, nrow=n, ncol=m,  byrow = FALSE)
rownames(dane)<-paste(rep("sector", times=n),1:n)
colnames(dane)<-paste(rep("region", times=m), 1:m)
sum.by.regions<-colSums(dane)
dane<-rbind(dane, sum.by.regions)
sum.by.sectors<-rowSums(dane)
dane<-cbind(dane, sum.by.sectors)
dane

empl<-dane[1:n, 1:m]
sectors<-rownames(dane[1:n,])

################################\
# data ranges for output
range.GiniGC<-1:n
range.KrugmanGC<-(n+1):(2*n)
range.BT<-(2*n+1):(3*n)
range.locGini<-(3*n+1):(4*n)
range.AgglomerationV<-(4*n+1):(5*n)
range.ShannonGC<-(5*n+1):(6*n)
range.RelativeHGC<-(6*n+1):(7*n)
range.TheilGC<-(7*n+1):(8*n)
range.KLDGC<-(8*n+1):(9*n)
################################\

################################\
# common elements of calculations
# matrix of industrial shares by regions
shares.in.rows<-matrix(0, ncol=m, nrow=n)
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 rows

# vector of industrial shares for national economy
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)}
shares.in.columns
colSums(shares.in.columns) # check of sums in columns

share.in.columns.extra<-sum.by.sectors/sum.by.sectors[n+1]
share.in.columns.extra

################################\
# Gini
wynik<-matrix(0, ncol=n, nrow=1)
colnames(wynik)<-paste("sector",1:n)
rownames(wynik)<-c("Gini")
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<-mean(vec3, na.rm=FALSE)
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
fin_obj[t,range.GiniGC]<-wynik

################################\
# Krugman concentration index
```

```
wynik<-matrix(0, ncol=n, nrow=1)
colnames(wynik)<-paste("sector",1:n)
rownames(wynik)<-c("Krugman Concentration")

for(i in 1:n){
diff.abs<-abs(shares.in.rows[i,]-share.in.rows.extra)
wynik[1,i]<-sum(diff.abs)}
wynik
fin_obj[t,range.KrugmanGC]<-wynik

################################\
# Bruelhart -Traeger index
wynik<-matrix(0, ncol=n, nrow=1)
colnames(wynik)<-paste("sector",1:n)
rownames(wynik)<-c("Bruelhart-Traeger")

temp<-matrix(0, nrow=n, ncol=m)
for(i in 1:n){
temp[i,]<-as.matrix(shares.in.rows[i,]/share.in.rows.extra)}

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)}
temp2

wynik<-rowSums(temp2)
wynik
fin_obj[t,range.BT]<-wynik

################################\
# locational Gini
wynik<-matrix(0, ncol=n, nrow=1)
colnames(wynik)<-paste("sector",1:n)
rownames(wynik)<-c("locGini")

temp<-matrix(0, nrow=n, ncol=m)
for(i in 1:n){
temp[i,]<-as.matrix(shares.in.columns[i,]/share.in.rows.extra)}
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
fin_obj[t,range.locGini]<-wynik
```

```
#################################\
# Agglomeration V
wynik<-matrix(0, ncol=n, nrow=1)
colnames(wynik)<-paste("sector",1:n)
rownames(wynik)<-c("Agglomeration V")

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-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
fin_obj[t,range.AgglomerationV]<-wynik

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

# 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")

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

fin_obj[t,range.ShannonGC]<-wynik[1,]
fin_obj[t,range.RelativeHGC]<-wynik[2,]
```

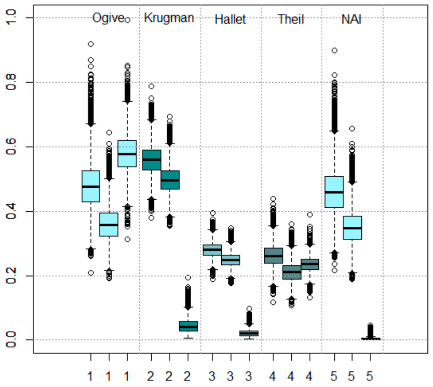
```
fin_obj[t,range.TheilGC]<-wynik[3,]
fin_obj[t,range.KLDGC]<-wynik[4,]

######
return(fin_obj)[t,]
} # closing the loop for the function
#################################\

# calling GCmeasures() function  “(GCmeasures<-function(dane, n, m, t, fin_obj))”

# big tables
fin_obj<-GC1
for(t in 1:1000){
GC1[t,]<-GCmeasures(get(paste("m1.dr.b.",t, sep="")), 100, 25, t, fin_obj=GC1)[t,]}
summary(GC1)

fin_obj<-GC2
for(t in 1:1000){
GC2[t,]<-GCmeasures(get(paste("m1.sr.b.",t, sep="")), 100, 25, t, fin_obj=GC2)[t,]}
summary(GC2)

fin_obj<-GC3
for(t in 1:1000){
GC3[t,]<-GCmeasures(get(paste("m1.p.b.",t, sep="")), 100, 25, t, fin_obj=GC3)[t,]}
summary(GC3)

fin_obj<-GC4
for(t in 1:1000){
GC4[t,]<-GCmeasures(get(paste("m2.dr.b.",t, sep="")), 100, 25, t, fin_obj=GC4)[t,]}
summary(GC4)

fin_obj<-GC5
for(t in 1:1000){
GC5[t,]<-GCmeasures(get(paste("m2.sr.b.",t, sep="")), 100, 25, t, fin_obj=GC5)[t,]}
summary(GC5)

fin_obj<-GC6
for(t in 1:1000){
GC6[t,]<-GCmeasures(get(paste("m2.p.b.",t, sep="")), 100, 25, t, fin_obj=GC6)[t,]}
summary(GC6)

# middle tables
fin_obj<-GC7
for(t in 1:1000){
GC7[t,]<-GCmeasures(get(paste("m1.dr.m.",t, sep="")), 25, 10, t, fin_obj=GC7)[t,]}
summary(GC7)

fin_obj<-GC8
for(t in 1:1000){
GC8[t,]<-GCmeasures(get(paste("m1.sr.m.",t, sep="")), 25, 10, t, fin_obj=GC8)[t,]}
summary(GC8)

fin_obj<-GC9
for(t in 1:1000){
GC9[t,]<-GCmeasures(get(paste("m1.p.m.",t, sep="")), 25, 10, t, fin_obj=GC9)[t,]}
summary(GC9)

fin_obj<-GC10
for(t in 1:1000){
GC10[t,]<-GCmeasures(get(paste("m2.dr.m.",t, sep="")), 25, 10, t, fin_obj=GC10)[t,]}
summary(GC10)

fin_obj<-GC11
for(t in 1:1000){
GC11[t,]<-GCmeasures(get(paste("m2.sr.m.",t, sep="")), 25, 10, t, fin_obj=GC11)[t,]}
```

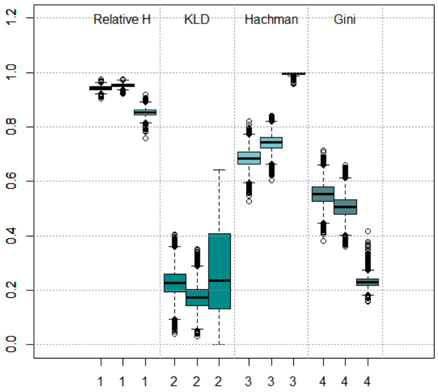
```
summary(GC11)

fin_obj<-GC12
for(t in 1:1000){
GC12[t,]<-GCmeasures(get(paste("m2.p.m.",t, sep="")), 25, 10, t, fin_obj=GC12)[t,]}
summary(GC12)

# small tables
fin_obj<-GC13
for(t in 1:1000){
GC13[t,]<-GCmeasures(get(paste("m1.dr.s.",t, sep="")), 5, 5, t, fin_obj=GC13)[t,]}
summary(GC13)

fin_obj<-GC14
for(t in 1:1000){
GC14[t,]<-GCmeasures(get(paste("m1.sr.s.",t, sep="")), 5, 5, t, fin_obj=GC14)[t,]}
summary(GC14)

fin_obj<-GC15
for(t in 1:1000){
GC15[t,]<-GCmeasures(get(paste("m1.p.s.",t, sep="")), 5, 5, t, fin_obj=GC15)[t,]}
summary(GC15)

fin_obj<-GC16
for(t in 1:1000){
GC16[t,]<-GCmeasures(get(paste("m2.dr.s.",t, sep="")), 5, 5, t, fin_obj=GC16)[t,]}
summary(GC16)

fin_obj<-GC17
for(t in 1:1000){
GC17[t,]<-GCmeasures(get(paste("m2.sr.s.",t, sep="")), 5, 5, t, fin_obj=GC17)[t,]}
summary(GC17)

fin_obj<-GC18
for(t in 1:1000){
GC18[t,]<-GCmeasures(get(paste("m2.p.s.",t, sep="")), 5, 5, t, fin_obj=GC18)[t,]}
summary(GC18)

# stack of GC measures matrix - one GC measure one column

size<-rep(c(25,10,5), each=6)

for(i in 1:18){
temp<-get(paste("GC", i, sep=""))
n<-size[i]

range.GiniGC<-1:n
s1<-temp[,range.GiniGC]
ss1<-stack(as.data.frame(s1))

range.KrugmanGC<-(n+1):(2*n)
s2<-temp[,range.KrugmanGC]
ss2<-stack(as.data.frame(s2))

range.BT<-(2*n+1):(3*n)
s3<-temp[,range.BT]
ss3<-stack(as.data.frame(s3))

range.locGini<-(3*n+1):(4*n)
s4<-temp[,range.locGini]
ss4<-stack(as.data.frame(s4))

range.AgglomerationV<-(4*n+1):(5*n)
s5<-temp[,range.AgglomerationV]
ss5<-stack(as.data.frame(s5))
range.ShannonGC<-(5*n+1):(6*n)
s6<-temp[,range.ShannonGC]
ss6<-stack(as.data.frame(s6))

range.RelativeHGC<-(6*n+1):(7*n)
s7<-temp[,range.RelativeHGC]
ss7<-stack(as.data.frame(s7))

range.TheilGC<-(7*n+1):(8*n)
s8<-temp[,range.TheilGC]
ss8<-stack(as.data.frame(s8))

range.KLDGC<-(8*n+1):(9*n)
s9<-temp[,range.KLDGC]
ss9<-stack(as.data.frame(s9))

temp.stack<-cbind(ss1$values, ss2$values, ss3$values, ss4$values, ss5$values,
ss6$values, ss7$values, ss8$values, ss9$values)

colnames(temp.stack)<-c("Gini", "Krugman Concentration", "Bruelhart-Traeger",
"locGini", "Agglomeration V", "Shannon’s H", "Relative H", "Theil’s H", "KLD")

assign(paste("GC", i, ".stack", sep=""), temp.stack)}
```

```
# RESULTS IN THE PAPER

# Tab.4A Statistics of variability of SC measures

śr1<-apply(SC1.stack, 2, mean)
round(śr1,2)
śr2<-apply(SC7.stack, 2, mean)
round(śr2,2)
śr3<-apply(SC13.stack, 2, mean)
round(śr3,2)

odch1<-apply(SC1.stack, 2, sd)
round(odch1,2)
odch2<-apply(SC7.stack, 2, sd)
round(odch2,2)
odch3<-apply(SC13.stack, 2, sd)
round(odch3,2)

round(odch1/śr1,2)
round(odch2/śr2,2)
round(odch3/śr3,2)

mmin<-apply(SC1.stack, 2, min)
round(mmin, 2)
mmin<-apply(SC7.stack, 2, min)
round(mmin, 2)
mmin<-apply(SC13.stack, 2, min)
round(mmin, 2)

mmax<-apply(SC1.stack, 2, max)
round(mmax, 2)
mmax<-apply(SC7.stack, 2, max)
round(mmax, 2)
mmax<-apply(SC13.stack, 2, max)
round(mmax, 2)


# Tab.4B Statistics of variability of GC measures
śr1<-apply(GC1.stack, 2, mean)
round(śr1,2)
śr2<-apply(GC7.stack, 2, mean)
round(śr2,2)
śr3<-apply(GC13.stack, 2, mean)
round(śr3,2)

odch1<-apply(GC1.stack, 2, sd)
round(odch1,2)
odch2<-apply(GC7.stack, 2, sd)
round(odch2,2)
odch3<-apply(GC13.stack, 2, sd)
round(odch3,2)

round(odch1/śr1,2)
round(odch2/śr2,2)
round(odch3/śr3,2)

mmin<-apply(GC1.stack, 2, min)
round(mmin, 2)
mmin<-apply(GC7.stack, 2, min)
round(mmin, 2)
mmin<-apply(GC13.stack, 2, min)
round(mmin, 2)
mmax<-apply(GC1.stack, 2, max)
round(mmax, 2)
mmax<-apply(GC7.stack, 2, max)
```

```
round(mmax, 2)
mmax<-apply(GC13.stack, 2, max)
round(mmax, 2)
```

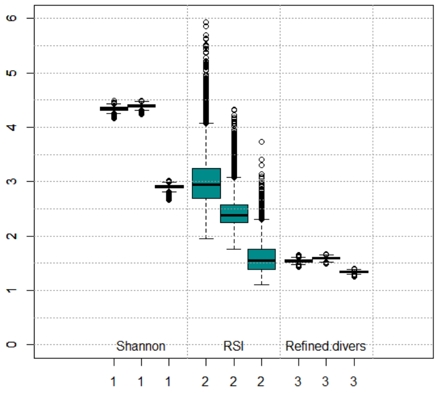
```
# FIGURE 2A: barplot of variability of SC measures

boxplot(SC1.stack[,c(1,2,4,8,3)], at=(1:5)-0.3, boxwex=0.3, names=1:5, ylim=c(0,1),
xlim=c(0,6), col=c("cadetblue1","darkcyan","cadetblue3","cadetblue4"))
boxplot(SC6.stack[,c(1,2,4,8,3)], at=(1:5), add=TRUE, boxwex=0.3, names=1:5,
col=c("cadetblue1","darkcyan","cadetblue3","cadetblue4"))
boxplot(SC13.stack[,c(1,2,4,8,3)], at=(1:5)+0.3, add=TRUE, boxwex=0.3, names=1:5,
col=c("cadetblue1","darkcyan","cadetblue3","cadetblue4"))
abline(h=c(0:5)*0.2, lty=3, col="grey60")
abline(v=c(1:5)+0.5, lty=3, col="grey60")
text(1,1, c("Ogive"))
text(2,1, c("Krugman"))
text(3,1, c("Hallet"))
text(4,1, c("Theil"))
text(5,1, c("NAI"))

boxplot(SC1.stack[,c(7,9,11,13)], at=(1:4)-0.3, boxwex=0.3, names=1:4, ylim=c(0,1.2),
xlim=c(0,5), col=c("cadetblue1","darkcyan","cadetblue3","cadetblue4"))
boxplot(SC6.stack[,c(7,9,11,13)], at=(1:4), add=TRUE, boxwex=0.3, names=1:4,
col=c("cadetblue1","darkcyan","cadetblue3","cadetblue4"))
boxplot(SC13.stack[,c(7,9,11,13)], at=(1:4)+0.3, add=TRUE, boxwex=0.3, names=1:4,
col=c("cadetblue1","darkcyan","cadetblue3","cadetblue4"))
abline(h=c(0:10)*0.2, lty=3, col="grey60")
abline(v=c(1:5)+0.5, lty=3, col="grey60")
text(1,1.2, c("Relative H"))
text(2,1.2, c("KLD"))
text(3,1.2, c("Hachman"))
text(4,1.2, c("Gini"))
```

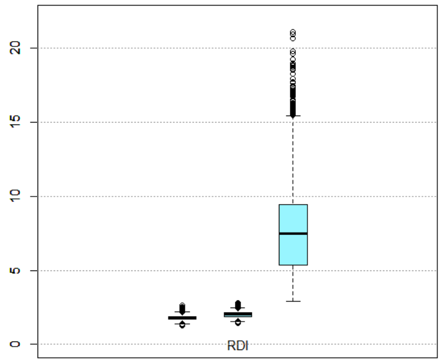




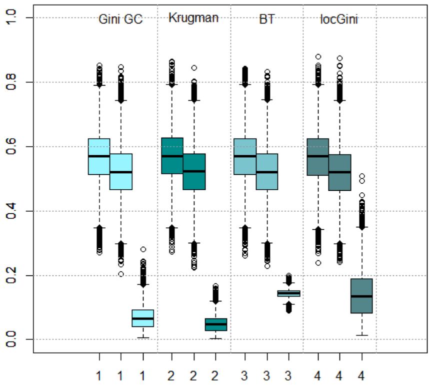
```
boxplot(SC1.stack[,c(6, 12, 10)], at=(1:3)-0.3, boxwex=0.3, names=1:3, ylim=c(0,6),
xlim=c(0,4), col=c("cadetblue1","darkcyan","cadetblue3","cadetblue4"))
boxplot(SC6.stack[,c(6,12, 10)], at=(1:3), add=TRUE, boxwex=0.3, names=1:3,
col=c("cadetblue1","darkcyan","cadetblue3","cadetblue4"))
boxplot(SC12.stack[,c(6,12,10)], at=(1:3)+0.3, add=TRUE, boxwex=0.3, names=1:3,
col=c("cadetblue1","darkcyan","cadetblue3","cadetblue4"))
abline(h=c(0:20)*0.5, lty=3, col="grey60")
abline(v=c(1:5)+0.5, lty=3, col="grey60")
text(1,0, c("Shannon"))
text(2,0, c("RSI"))
text(3,0, c("Refined.divers"))

boxplot(SC1.stack[,c(5)], at=(1:1)-0.3, boxwex=0.3, names=1:1, ylim=c(0,22),
xlim=c(0,2), col=c("cadetblue1","darkcyan","cadetblue3","cadetblue4"))
boxplot(SC6.stack[,c(5)], at=(1:1), add=TRUE, boxwex=0.3, names=1:1,
col=c("cadetblue1","darkcyan","cadetblue3","cadetblue4"))
boxplot(SC12.stack[,c(5)], at=(1:1)+0.3, add=TRUE, boxwex=0.3, names=1:1,
col=c("cadetblue1","darkcyan","cadetblue3","cadetblue4"))
text(1,0, c("RDI"))
```

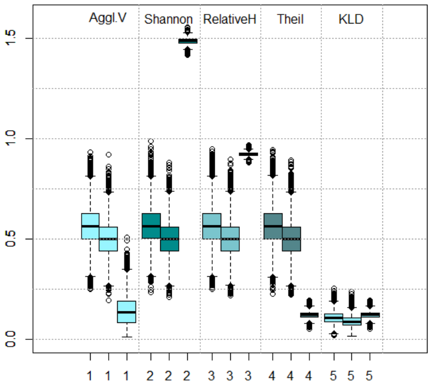
```
abline(h=c(0:25)*5, lty=3, col="grey60")
```



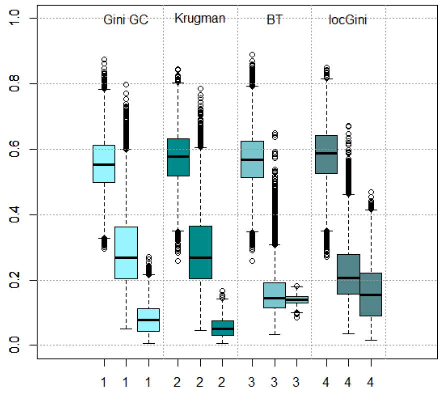


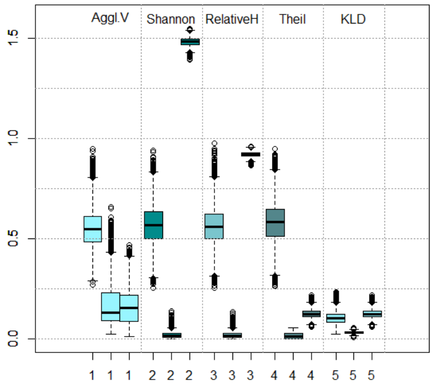
```
# FIGURE 2B: barplot of variability of GC measures

# for tables 1, 6 & 13
boxplot(GC1.stack[,c(1,2,3,4)], at=(1:4)-0.3, boxwex=0.3, names=1:4, ylim=c(0,1),
xlim=c(0,5), col=c("cadetblue1","darkcyan","cadetblue3","cadetblue4"))
boxplot(GC6.stack[,c(1,2,3,4)], at=(1:4), add=TRUE, boxwex=0.3, names=1:4,
col=c("cadetblue1","darkcyan","cadetblue3","cadetblue4"))
boxplot(GC13.stack[,c(1,2,4,5)], at=(1:4)+0.3, add=TRUE, boxwex=0.3, names=1:4,
col=c("cadetblue1","darkcyan","cadetblue3","cadetblue4"))
abline(h=c(0:5)*0.2, lty=3, col="grey60")
abline(v=c(1:5)+0.5, lty=3, col="grey60")
text(1,1, c("Gini GC"))
text(2,1, c("Krugman"))
text(3,1, c("BT"))
text(4,1, c("locGini"))

boxplot(GC1.stack[,c(5,6,7,8,9)], at=(1:5)-0.3, boxwex=0.3, names=1:5, ylim=c(0,1.6),
xlim=c(0,6), col=c("cadetblue1","darkcyan","cadetblue3","cadetblue4", "cadetblue2"))
boxplot(GC6.stack[,c(5,6,7,8,9)], at=(1:5), add=TRUE, boxwex=0.3, names=1:5,
col=c("cadetblue1","darkcyan","cadetblue3","cadetblue4", "cadetblue2"))
boxplot(GC13.stack[,c(5,6,7,8,9)], at=(1:5)+0.3, add=TRUE, boxwex=0.3, names=1:5,
col=c("cadetblue1","darkcyan","cadetblue3","cadetblue4", "cadetblue2"))
abline(h=c(0:10)*0.25, lty=3, col="grey60")
abline(v=c(1:5)+0.5, lty=3, col="grey60")
text(1,1.6, c("Aggl.V"))
text(2,1.6, c("Shannon"))
text(3,1.6, c("RelativeH"))
text(4,1.6, c("Theil"))
text(5,1.6, c("KLD"))
```



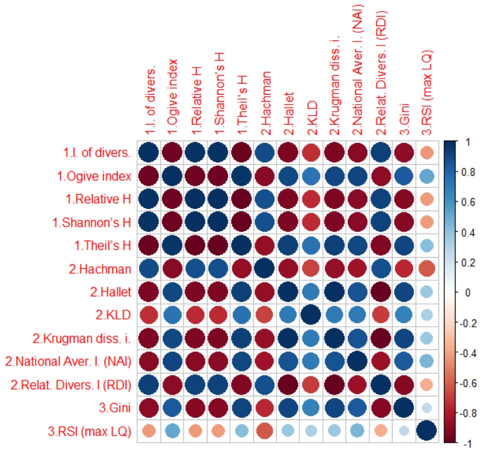
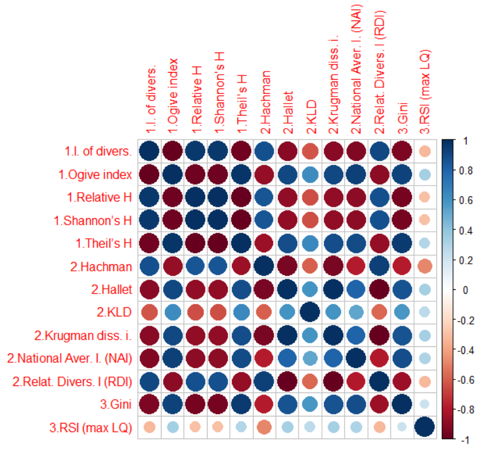


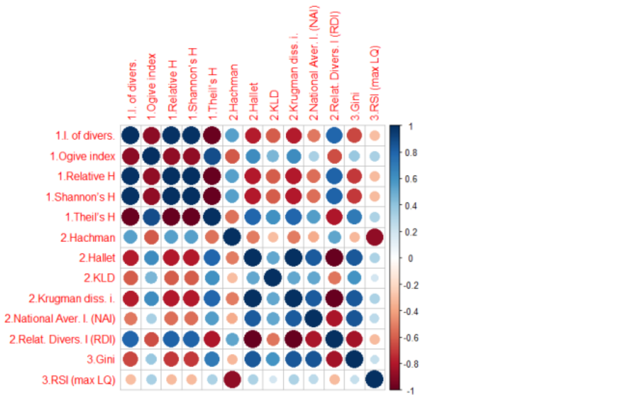
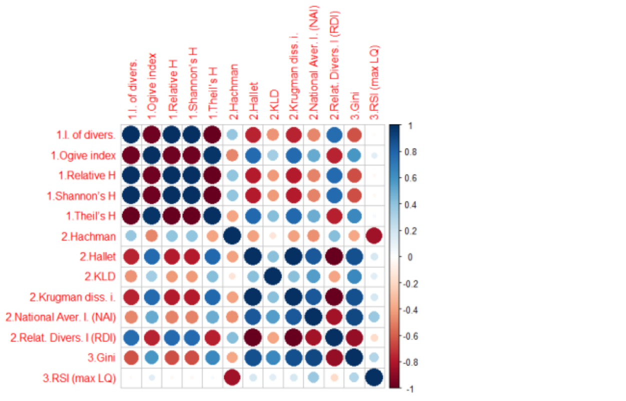
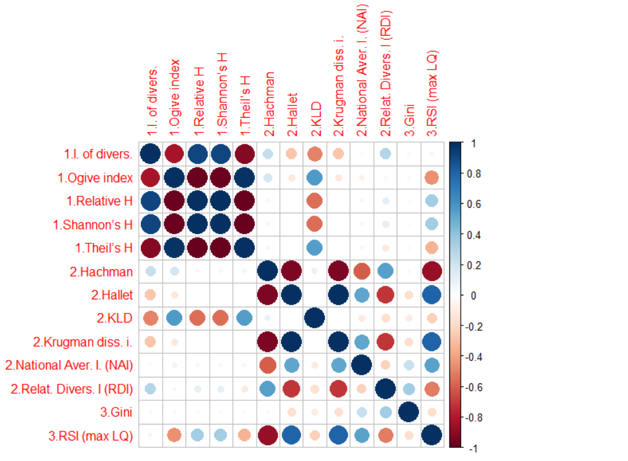
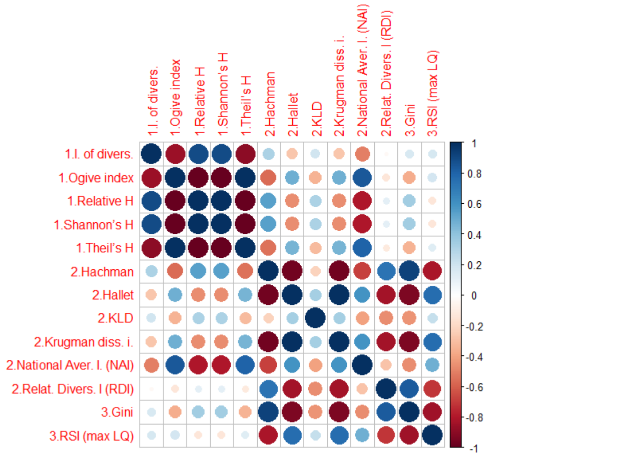
```
# for tables 2, 7 & 14
boxplot(GC2.stack[,c(1,2,3,4)], at=(1:4)-0.3, boxwex=0.3, names=1:4, ylim=c(0,1),
xlim=c(0,5), col=c("cadetblue1","darkcyan","cadetblue3","cadetblue4"))
boxplot(GC7.stack[,c(1,2,3,4)], at=(1:4), add=TRUE, boxwex=0.3, names=1:4,
col=c("cadetblue1","darkcyan","cadetblue3","cadetblue4"))
boxplot(GC14.stack[,c(1,2,4,5)], at=(1:4)+0.3, add=TRUE, boxwex=0.3, names=1:4,
col=c("cadetblue1","darkcyan","cadetblue3","cadetblue4"))
abline(h=c(0:5)*0.2, lty=3, col="grey60")
abline(v=c(1:5)+0.5, lty=3, col="grey60")
text(1,1, c("Gini GC"))
text(2,1, c("Krugman"))
text(3,1, c("BT"))
text(4,1, c("locGini"))

boxplot(GC2.stack[,c(5,6,7,8,9)], at=(1:5)-0.3, boxwex=0.3, names=1:5, ylim=c(0,1.6),
xlim=c(0,6), col=c("cadetblue1","darkcyan","cadetblue3","cadetblue4", "cadetblue2"))
boxplot(GC7.stack[,c(5,6,7,8,9)], at=(1:5), add=TRUE, boxwex=0.3, names=1:5,
col=c("cadetblue1","darkcyan","cadetblue3","cadetblue4", "cadetblue2"))
boxplot(GC14.stack[,c(5,6,7,8,9)], at=(1:5)+0.3, add=TRUE, boxwex=0.3, names=1:5,
col=c("cadetblue1","darkcyan","cadetblue3","cadetblue4", "cadetblue2"))
abline(h=c(0:10)*0.25, lty=3, col="grey60")
abline(v=c(1:5)+0.5, lty=3, col="grey60")
text(1,1.6, c("Aggl.V"))
text(2,1.6, c("Shannon"))
text(3,1.6, c("RelativeH"))
text(4,1.6, c("Theil"))
text(5,1.6, c("KLD"))
```





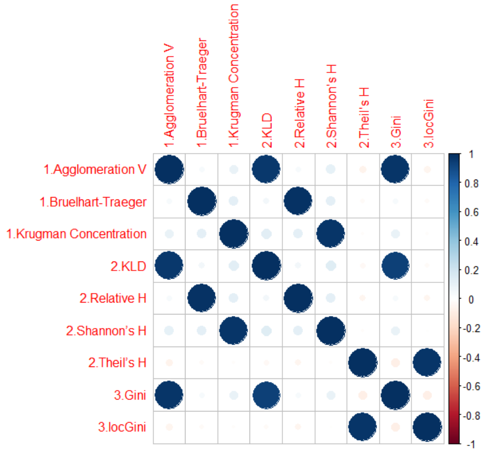
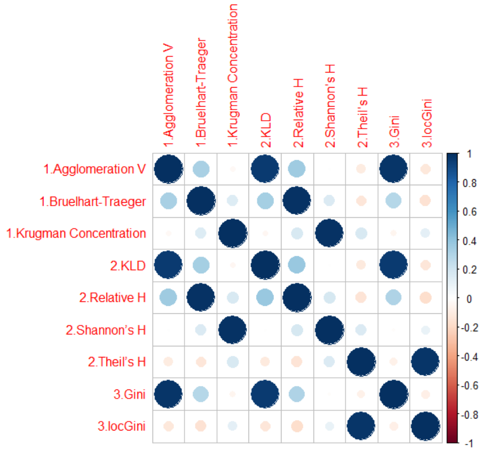
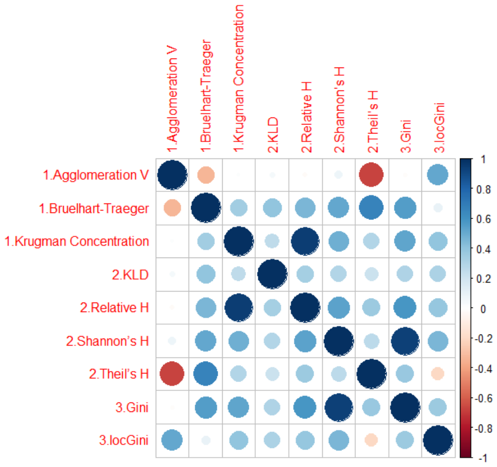
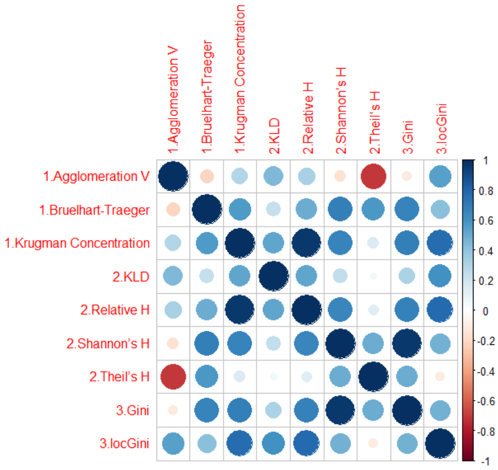
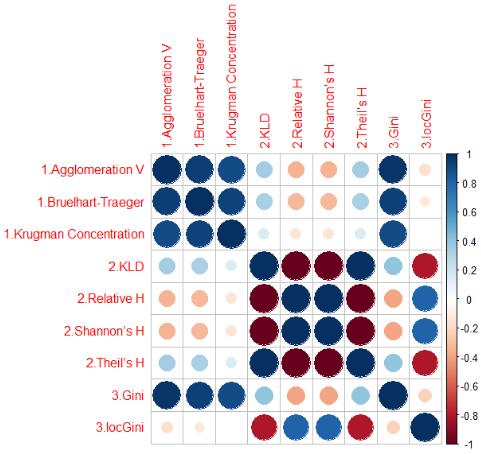
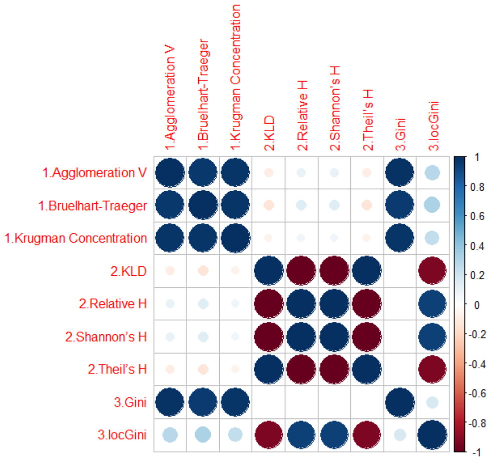
```
# Table 5A: correlations of SC measures - with numbers in names

# Single randomisation
library(corrplot)

colnames(SC2.stack)<-c("1.Ogive index", "2.Krugman diss. i.", "2.National Aver. I.
(NAI)", "2.Hallet", "2.Relat. Divers. I (RDI)", "1.Shannon’s H", "1.Relative H",
"1.Theil’s H", "2.KLD", "1.I. of divers.", "2.Hachman", "3.RSI (max LQ)", "3.Gini")
korSC2.p<-cor(SC2.stack, method="pearson")
corrplot(korSC2.p, order ="alphabet")

colnames(SC5.stack)<-c("1.Ogive index", "2.Krugman diss. i.", "2.National Aver. I.
(NAI)", "2.Hallet", "2.Relat. Divers. I (RDI)", "1.Shannon’s H", "1.Relative H",
"1.Theil’s H", "2.KLD", "1.I. of divers.", "2.Hachman", "3.RSI (max LQ)", "3.Gini")
korSC2.p<-cor(SC5.stack, method="pearson")
corrplot(korSC2.p, order ="alphabet")

colnames(SC8.stack)<-c("1.Ogive index", "2.Krugman diss. i.", "2.National Aver. I.
(NAI)", "2.Hallet", "2.Relat. Divers. I (RDI)", "1.Shannon’s H", "1.Relative H",
"1.Theil’s H", "2.KLD", "1.I. of divers.", "2.Hachman", "3.RSI (max LQ)", "3.Gini")
korSC2.p<-cor(SC8.stack, method="pearson")
corrplot(korSC2.p, order ="alphabet")

colnames(SC11.stack)<-c("1.Ogive index", "2.Krugman diss. i.", "2.National Aver. I.
(NAI)", "2.Hallet", "2.Relat. Divers. I (RDI)", "1.Shannon’s H", "1.Relative H",
"1.Theil’s H", "2.KLD", "1.I. of divers.", "2.Hachman", "3.RSI (max LQ)", "3.Gini")
korSC2.p<-cor(SC11.stack, method="pearson")
corrplot(korSC2.p, order ="alphabet")

colnames(SC14.stack)<-c("1.Ogive index", "2.Krugman diss. i.", "2.National Aver. I.
(NAI)", "2.Hallet", "2.Relat. Divers. I (RDI)", "1.Shannon’s H", "1.Relative H",
"1.Theil’s H", "2.KLD", "1.I. of divers.", "2.Hachman", "3.RSI (max LQ)", "3.Gini")
korSC2.p<-cor(SC14.stack, method="pearson")
corrplot(korSC2.p, order ="alphabet")

colnames(SC17.stack)<-c("1.Ogive index", "2.Krugman diss. i.", "2.National Aver. I.
(NAI)", "2.Hallet", "2.Relat. Divers. I (RDI)", "1.Shannon’s H", "1.Relative H",
"1.Theil’s H", "2.KLD", "1.I. of divers.", "2.Hachman", "3.RSI (max LQ)", "3.Gini")
korSC2.p<-cor(SC17.stack, method="pearson")
corrplot(korSC2.p, order ="alphabet")
```

	N1	N2
Big table		

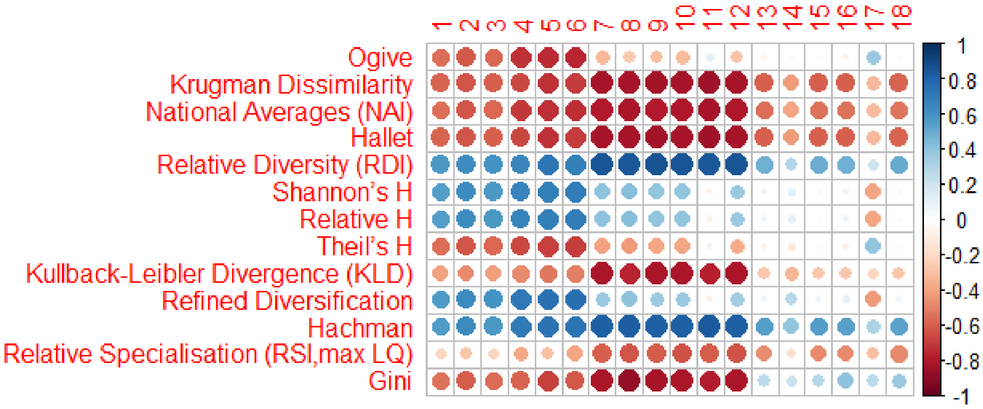
Middle table		
Small table		

```
# Table 5B: correlations of GC measures – with numbers in names

# Single randomisation
library(corrplot)
colnames(GC2.stack)<-c("3.Gini", "1.Krugman Concentration", "1.Bruelhart-Traeger",
"3.locGini", "1.Agglomeration V", "2.Shannon’s H", "2.Relative H", "2.Theil’s H",
"2.KLD")
korGC2.p<-cor(GC2.stack, method="pearson")
corrplot(korGC2.p, order ="alphabet")

colnames(GC5.stack)<-c("3.Gini", "1.Krugman Concentration", "1.Bruelhart-Traeger",
"3.locGini", "1.Agglomeration V", "2.Shannon’s H", "2.Relative H", "2.Theil’s H",
"2.KLD")
korGC2.p<-cor(GC5.stack, method="pearson")
corrplot(korGC2.p, order ="alphabet")

colnames(GC8.stack)<-c("3.Gini", "1.Krugman Concentration", "1.Bruelhart-Traeger",
"3.locGini", "1.Agglomeration V", "2.Shannon’s H", "2.Relative H", "2.Theil’s H",
"2.KLD")
korGC2.p<-cor(GC8.stack, method="pearson")
corrplot(korGC2.p, order ="alphabet")

colnames(GC11.stack)<-c("3.Gini", "1.Krugman Concentration", "1.Bruelhart-Traeger",
"3.locGini", "1.Agglomeration V", "2.Shannon’s H", "2.Relative H", "2.Theil’s H",
"2.KLD")
korGC2.p<-cor(GC11.stack, method="pearson")
corrplot(korGC2.p, order ="alphabet")

colnames(GC14.stack)<-c("3.Gini", "1.Krugman Concentration", "1.Bruelhart-Traeger",
"3.locGini", "1.Agglomeration V", "2.Shannon’s H", "2.Relative H", "2.Theil’s H",
"2.KLD")
korGC2.p<-cor(GC14.stack, method="pearson")
corrplot(korGC2.p, order ="alphabet")
```

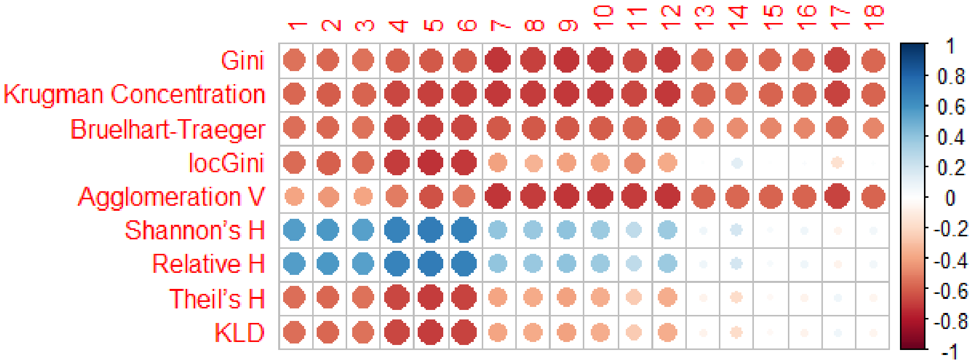
```
colnames(GC17.stack)<-c("3.Gini", "1.Krugman Concentration", "1.Bruelhart-Traeger",
"3.locGini", "1.Agglomeration V", "2.Shannon's H", "2.Relative H", "2.Theil's H",
"2.KLD")
korGC2.p<-cor(GC17.stack, method="pearson")
corrplot(korGC2.p, order ="alphabet")
```

	N1	N2
Big table		
Middle table		
Small table		

```
# FIG.3A, FIG.4, FIG.5 - big loop to calculate correlations between SC measures and
size of the region

m.size<-rep(c(25,10,5), each=6)
macierz<-c("m1.dr.b.", "m1.sr.b.", "m1.p.b.", "m2.dr.b.", "m2.sr.b.", "m2.p.b.",
      "m1.dr.m.", "m1.sr.m.", "m1.p.m.", "m2.dr.m.", "m2.sr.m.", "m2.p.m.",
      "m1.dr.s.", "m1.sr.s.", "m1.p.s.", "m2.dr.s.", "m2.sr.s.", "m2.p.s.")
miara<-c("Ogive", "Krugman Dissimilarity", "National Averages (NAI)", "Hallet",
"Relative Diversity (RDI)", "Shannon’s H", "Relative H", "Theil’s H", "Kullback-Leibler
Divergence (KLD)", "Refined Diversification", "Hachman", "Relative Specialisation
(RSI,max LQ)", "Gini")
range<-c("range.Ogive", "range.Krugman", "range.NAI", "range.Hallet", "range.RDI",
"range.Shannon", "range.RelativeH", "range.Theil", "range.KLD", "range.divers",
"range.Hachman", "range.RSI", "range.Gini")

big.matrix.kor<-matrix(0, nrow=18, ncol=13)
big.matrix.pval<-matrix(0, nrow=18, ncol=13)

for(i in 1:18){ # wybór schematu  1:18
for(j in 1:13){ # wybór miary 1:13

m<-m.size[i]

range.Ogive<-1:m
range.Krugman<-(m+1):(2*m)
range.NAI<-(2*m+1):(3*m)
range.Hallet<-(3*m+1):(4*m)
range.RDI<-(4*m+1):(5*m)
range.Shannon<-(5*m+1):(6*m)
range.RelativeH<-(6*m+1):(7*m)
range.Theil<-(7*m+1):(8*m)
range.KLD<-(8*m+1):(9*m)
range.divers<-(9*m+1):(10*m)
range.Hachman<-(10*m+1):(11*m)
range.RSI<-(11*m+1):(12*m)
range.Gini<-(12*m+1):(13*m)

SC.temp<-get(paste("SC", i, sep=""))
miara.temp<-miara[j]
range.temp<-get(range[j])

dane.y<-SC.temp[, range.temp]
dane.y.stack<-stack(as.data.frame(dane.y))

dane.x<-matrix(0, nrow=1000, ncol=m)
for(t in 1:1000){ # wybór iteracji
macierz.temp<-get(paste(macierz[i], t, sep=""))
vec<-colSums(macierz.temp)
vec1<-vec/sum(vec)
dane.x[t,]<-vec1}
dane.x.stack<-stack(as.data.frame(dane.x))

kore<-cor.test(dane.x.stack$values, dane.y.stack$values)
kore
plot(dane.x, dane.y, pch=".", xlab="share of regional employment in overall economy",
ylab=miara.temp, main=paste(miara.temp, "     m=", m, "      ", macierz[i], sep=""))
text(min(dane.x), min(dane.y), labels=paste("corr = ", round(kore$estimate,2), "
(p-value=", round(kore$p.value,2), ")", sep=""), pos=4)
savePlot(filename=paste(miara.temp, "SC_", i, sep=""), type="jpg")
big.matrix.kor[i,j]<-kore$estimate
big.matrix.pval[i,j]<-kore$p.value}}

colnames(big.matrix.kor)<-miara
corrplot(t(big.matrix.kor))
```


1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
Ogive
Krugman Dissimilarity
National Averages (NAI)
Hallet
Relative Diversity (RDI)
Shannon's H
Relative H
Theil's H
Kullback-Leibler Divergence (KLD)
Refined Diversification
Hachman
Relative Specialisation (RSI,max LQ)
Gini
1
0.8
0.6
0.4
0.2
0
-0.2
-0.4
-0.6
-0.8
-1

```
# FIG.3B big loop to calculate correlations between GC measures and size of the region

n.size<-rep(c(100,25,5), each=6)
macierz<-c("m1.dr.b.", "m1.sr.b.", "m1.p.b.", "m2.dr.b.", "m2.sr.b.", "m2.p.b.",
      "m1.dr.m.", "m1.sr.m.", "m1.p.m.", "m2.dr.m.", "m2.sr.m.", "m2.p.m.",
      "m1.dr.s.", "m1.sr.s.", "m1.p.s.", "m2.dr.s.", "m2.sr.s.", "m2.p.s.")
miara<- c("Gini", "Krugman Concentration", "Bruelhart-Traeger", "locGini",
"Agglomeration V", "Shannon’s H", "Relative H", "Theil’s H", "KLD")

range<-c("range.GiniGC", "range.KrugmanGC", "range.BT", "range.locGini",
"range.AgglomerationV", "range.ShannonGC", "range.RelativeHGC", "range.TheilGC",
"range.KLDGC")

big.matrix.kor<-matrix(0, nrow=18, ncol=9)
big.matrix.pval<-matrix(0, nrow=18, ncol=9)

for(i in 1:18){ # wybór schematu  1:18
for(j in 1:9){ # wybór miary 1:9

n<-n.size[i]

# data ranges for output
range.GiniGC<-1:n
range.KrugmanGC<-(n+1):(2*n)
range.BT<-(2*n+1):(3*n)
range.locGini<-(3*n+1):(4*n)
range.AgglomerationV<-(4*n+1):(5*n)
range.ShannonGC<-(5*n+1):(6*n)
range.RelativeHGC<-(6*n+1):(7*n)
range.TheilGC<-(7*n+1):(8*n)
range.KLDGC<-(8*n+1):(9*n)

GC.temp<-get(paste("GC", i, sep=""))
miara.temp<-miara[j]
range.temp<-get(range[j])

dane.y<-GC.temp[, range.temp]
dane.y.stack<-stack(as.data.frame(dane.y))

dane.x<-matrix(0, nrow=1000, ncol=n)
for(t in 1:1000){ # wybór iteracji
macierz.temp<-get(paste(macierz[i], t, sep=""))
vec<-rowSums(macierz.temp)
vec1<-vec/sum(vec)
dane.x[t,]<-vec1}
dane.x.stack<-stack(as.data.frame(dane.x))

kore<-cor.test(dane.x.stack$values, dane.y.stack$values)
kore
plot(dane.x, dane.y, pch=".", xlab="share of regional employment in overall economy",
ylab=miara.temp, main=paste(miara.temp, "     m=", m, "     ", macierz[i], sep=""))
text(min(dane.x), min(dane.y), labels=paste("corr = ", round(kore$estimate,2), "
(p-value=", round(kore$p.value,2), ")", sep=""), pos=4)
savePlot(filename=paste(miara.temp, "GC_", i, sep=""), type="jpg")
big.matrix.kor[i,j]<-kore$estimate
big.matrix.pval[i,j]<-kore$p.value}}

colnames(big.matrix.kor)<-miara
corrplot(t(big.matrix.kor))
```


1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
Gini
Krugman Concentration
Bruelhart-Traeger
locGini
Agglomeration V
Shannon’s H
Relative H
Theil’s H
KLD
1
0.8
0.6
0.4
0.2
0
-0.2
-0.4
-0.6
-0.8
-1

# TAB.6A&7A – Mantel test for matrices correlation

***H1: sensitivity to MAUP*. This is verified by comparing the measures for different degrees of aggregation (big, middle and small tables) and for selected distribution and sampling. This results in comparison of schemes (1,7,13), (2,8,14), (3,9,15), (4,10,16), (5,11,17), (6,12,18)**

```
library(vegan)
library(corrplot)

# comparison of correlation matrices for big, middle and small tables
wynik.mantel<-matrix(0, nrow=6, ncol=2)
rownames(wynik.mantel)<-c("N1 DR (1,7,13)","N1 SR (2,8,14)","N1 P (3,9,15)","N2 DR
(4,10,16)","N2 SR (5,11,17)","N2 P (6,12,18)")
colnames(wynik.mantel)<-c("Mantel z","p-value")

for(i in 1:6){
# macierze
a<-cor(get(paste("SC",i, ".stack", sep="")))
b<-cor(get(paste("SC",i+6, ".stack", sep="")))
c<-cor(get(paste("SC",i+12, ".stack", sep="")))

a<-mantel.partial(a, b, c, method = "pearson", permutations = 999)
wynik.mantel[i,1]<-a$statistic
wynik.mantel[i,2]<-a$signif}
wynik.mantel

# comparison of correlation matrices for big and small tables (no middle tables)
wynik.mantel<-matrix(0, nrow=6, ncol=2)
rownames(wynik.mantel)<-c("N1 DR (1,13)","N1 SR (2,14)","N1 P (3,15)","N2 DR
(4,16)","N2 SR (5,17)","N2 P (6,18)")
colnames(wynik.mantel)<-c("Mantel z","p-value")

for(i in 1:6){
# macierze
a<-cor(get(paste("SC",i, ".stack", sep="")))
c<-cor(get(paste("SC",i+12, ".stack", sep="")))

a<-mantel(a, c, method = "pearson", permutations = 999)
wynik.mantel[i,1]<-a$statistic
wynik.mantel[i,2]<-a$signif}
wynik.mantel

# comparison of distance matrices for big and small tables (no middle tables)
wynik.mantel<-matrix(0, nrow=6, ncol=2)
rownames(wynik.mantel)<-c("N1 DR (1,13)","N1 SR (2,14)","N1 P (3,15)","N2 DR
(4,16)","N2 SR (5,17)","N2 P (6,18)")
colnames(wynik.mantel)<-c("Mantel z","p-value")

for(i in 1:6){
# macierze
a<-dist(t(get(paste("SC",i, ".stack", sep=""))), method="euclidean")
c<-dist(t(get(paste("SC",i+12, ".stack", sep=""))), method="euclidean")

a<-mantel(a, c, method = "pearson", permutations = 999)
wynik.mantel[i,1]<-a$statistic
wynik.mantel[i,2]<-a$signif}
wynik.mantel


# comparison of distance matrices for big, middle and small tables
wynik.mantel<-matrix(0, nrow=6, ncol=2)
rownames(wynik.mantel)<-c("N1 DR (1,7,13)","N1 SR (2,8,14)","N1 P (3,9,15)","N2 DR
(4,10,16)","N2 SR (5,11,17)","N2 P (6,12,18)")
```

```
colnames(wynik.mantel)<-c("Mantel z","p-value")

for(i in 1:6){
# macierze
a<-dist(t(get(paste("SC",i, ".stack", sep=""))), method="euclidean")
b<-dist(t(get(paste("SC",i+6, ".stack", sep=""))), method="euclidean")
c<-dist(t(get(paste("SC",i+12, ".stack", sep=""))), method="euclidean")

a<-mantel.partial(a, b, c, method = "pearson", permutations = 999)
wynik.mantel[i,1]<-a$statistic
wynik.mantel[i,2]<-a$signif}
wynik.mantel
```

***H2: sensitivity of scale and variation of the input data*. This is verified by observing structural changes between measures, reflected by changes in pair-correlations and matrix correlations. This results in comparison of schemes (1,4), (2,5), and consequently (7,10), (8,11), and (13,16), (14,17)**

```
# for corr() function
wynik.mantel<-matrix(0, nrow=6, ncol=2)
rownames(wynik.mantel)<-c("Big DR N1, N2","Big SR N1 N2","Middle DR N1 N2","Middle SR
N1 N2","Small DR N1 N2","Small SR N1 N2")
colnames(wynik.mantel)<-c("Mantel z","p-value")

for(i in 1:6){
indeks1<-c(1,2,7,8,13,14)
indeks2<-c(4,5,10,11,16,17)

# macierze
a<-cor(get(paste("SC",indeks1[i], ".stack", sep="")))
b<-cor(get(paste("SC",indeks2[i], ".stack", sep="")))

a<-mantel(a, b, method = "pearson", permutations = 999)
wynik.mantel[i,1]<-a$statistic
wynik.mantel[i,2]<-a$signif}
wynik.mantel

# for dist() function
wynik.mantel<-matrix(0, nrow=6, ncol=2)
rownames(wynik.mantel)<-c("Big DR N1, N2","Big SR N1 N2","Middle DR N1 N2","Middle SR
N1 N2","Small DR N1 N2","Small SR N1 N2")
colnames(wynik.mantel)<-c("Mantel z","p-value")

for(i in 1:6){
indeks1<-c(1,2,7,8,13,14)
indeks2<-c(4,5,10,11,16,17)

# macierze
a<-dist(t(get(paste("SC",indeks1[i], ".stack", sep=""))), method="euclidean")
b<-dist(t(get(paste("SC",indeks2[i], ".stack", sep=""))), method="euclidean")

a<-mantel(a, b, method = "pearson", permutations = 999)
wynik.mantel[i,1]<-a$statistic
wynik.mantel[i,2]<-a$signif}
wynik.mantel
```

***H3: location invariant measures*. This is verified by observing the distributions of the measures and structural changes between measures, reflected by changes in pair-correlations and matrix correlations. This results in comparison of schemes (3,6), (9,12) and (15,18)**

```
# for corr() function
wynik.mantel<-matrix(0, nrow=3, ncol=2)
```

```
rownames(wynik.mantel)<-c("Big P N1, N2", "Middle P N1 N2","Small P N1 N2")
colnames(wynik.mantel)<-c("Mantel z","p-value")

for(i in 1:3){
indeks1<-c(3,9,15)
indeks2<-c(6,12,18)

# macierze
a<-cor(get(paste("SC",indeks1[i], ".stack", sep="")))
b<-cor(get(paste("SC",indeks2[i], ".stack", sep="")))

a<-mantel(a, b, method = "pearson", permutations = 999)
wynik.mantel[i,1]<-a$statistic
wynik.mantel[i,2]<-a$signif}
wynik.mantel


# for dist() function
wynik.mantel<-matrix(0, nrow=3, ncol=2)
rownames(wynik.mantel)<-c("Big P N1, N2", "Middle P N1 N2","Small P N1 N2")
colnames(wynik.mantel)<-c("Mantel z","p-value")

for(i in 1:3){
indeks1<-c(3,9,15)
indeks2<-c(6,12,18)

# macierze
a<-dist(t(get(paste("SC",indeks1[i], ".stack", sep=""))), method="euclidean")
b<-dist(t(get(paste("SC",indeks2[i], ".stack", sep=""))), method="euclidean")

a<-mantel(a, b, method = "pearson", permutations = 999)
wynik.mantel[i,1]<-a$statistic
wynik.mantel[i,2]<-a$signif}
wynik.mantel



# TAB.6B&7B - Mantel test for measures correlation
```

***H1: sensitivity to MAUP*. This is verified by comparing the measures for different degrees of aggregation (big, middle and small tables) and for selected distribution and sampling. This results in comparison of schemes (1,7,13), (2,8,14), (3,9,15), (4,10,16), (5,11,17), (6,12,18)**

```
library(vegan)
library(corrplot)

# comparison of correlation matrices for big, middle and small tables
wynik.mantel<-matrix(0, nrow=6, ncol=2)
rownames(wynik.mantel)<-c("N1 DR (1,7,13)","N1 SR (2,8,14)","N1 P (3,9,15)","N2 DR
(4,10,16)","N2 SR (5,11,17)","N2 P (6,12,18)")
colnames(wynik.mantel)<-c("Mantel z","p-value")

for(i in 1:6){
# macierze
a<-cor(get(paste("GC",i, ".stack", sep="")))
b<-cor(get(paste("GC",i+6, ".stack", sep="")))
c<-cor(get(paste("GC",i+12, ".stack", sep="")))

a<-mantel.partial(a, b, c, method = "pearson", permutations = 999)
wynik.mantel[i,1]<-a$statistic
wynik.mantel[i,2]<-a$signif}
wynik.mantel

# comparison of correlation matrices for big and small tables (no middle tables)
```

```
wynik.mantel<-matrix(0, nrow=6, ncol=2)
rownames(wynik.mantel)<-c("N1 DR (1,13)","N1 SR (2,14)","N1 P (3,15)","N2 DR
(4,16)","N2 SR (5,17)","N2 P (6,18)")
colnames(wynik.mantel)<-c("Mantel z","p-value")

for(i in 1:6){
# macierze
a<-cor(get(paste("GC",i, ".stack", sep="")))
c<-cor(get(paste("GC",i+12, ".stack", sep="")))

a<-mantel(a, c, method = "pearson", permutations = 999)
wynik.mantel[i,1]<-a$statistic
wynik.mantel[i,2]<-a$signif}
wynik.mantel

# comparison of distance matrices for big and small tables (no middle tables)
wynik.mantel<-matrix(0, nrow=6, ncol=2)
rownames(wynik.mantel)<-c("N1 DR (1,13)","N1 SR (2,14)","N1 P (3,15)","N2 DR
(4,16)","N2 SR (5,17)","N2 P (6,18)")
colnames(wynik.mantel)<-c("Mantel z","p-value")

for(i in 1:6){
# macierze
a<-dist(t(get(paste("GC",i, ".stack", sep=""))), method="euclidean")
c<-dist(t(get(paste("GC",i+12, ".stack", sep=""))), method="euclidean")

a<-mantel(a, c, method = "pearson", permutations = 999)
wynik.mantel[i,1]<-a$statistic
wynik.mantel[i,2]<-a$signif}
wynik.mantel

# comparison of distance matrices for big, middle and small tables
wynik.mantel<-matrix(0, nrow=6, ncol=2)
rownames(wynik.mantel)<-c("N1 DR (1,7,13)","N1 SR (2,8,14)","N1 P (3,9,15)","N2 DR
(4,10,16)","N2 SR (5,11,17)","N2 P (6,12,18)")
colnames(wynik.mantel)<-c("Mantel z","p-value")

for(i in 1:6){
# macierze
a<-dist(t(get(paste("GC",i, ".stack", sep=""))), method="euclidean")
b<-dist(t(get(paste("GC",i+6, ".stack", sep=""))), method="euclidean")
c<-dist(t(get(paste("GC",i+12, ".stack", sep=""))), method="euclidean")

a<-mantel.partial(a, b, c, method = "pearson", permutations = 999)
wynik.mantel[i,1]<-a$statistic
wynik.mantel[i,2]<-a$signif}
wynik.mantel
```

***H2: sensitivity of scale and variation of the input data*. This is verified by observing structural changes between measures, reflected by changes in pair-correlations and matrix correlations. This results in comparison of schemes (1,4), (2,5), and consequently (7,10), (8,11), and (13,16), (14,17).**

```
# for corr()
wynik.mantel<-matrix(0, nrow=6, ncol=2)
rownames(wynik.mantel)<-c("Big DR N1, N2","Big SR N1 N2","Middle DR N1 N2","Middle SR
N1 N2","Small DR N1 N2","Small SR N1 N2")
colnames(wynik.mantel)<-c("Mantel z","p-value")

for(i in 1:6){
indeks1<-c(1,2,7,8,13,14)
indeks2<-c(4,5,10,11,16,17)

a<-cor(get(paste("GC",indeks1[i], ".stack", sep="")))
b<-cor(get(paste("GC",indeks2[i], ".stack", sep="")))

a<-mantel(a, b, method = "pearson", permutations = 999)
wynik.mantel[i,1]<-a$statistic
wynik.mantel[i,2]<-a$signif}
wynik.mantel
```

# for dist()

```
wynik.mantel<-matrix(0, nrow=6, ncol=2)
rownames(wynik.mantel)<-c("Big DR N1, N2","Big SR N1 N2","Middle DR N1 N2","Middle SR
N1 N2","Small DR N1 N2","Small SR N1 N2")
colnames(wynik.mantel)<-c("Mantel z","p-value")

for(i in 1:6){
indeks1<-c(1,2,7,8,13,14)
indeks2<-c(4,5,10,11,16,17)

a<-dist(t(get(paste("GC",indeks1[i], ".stack", sep=""))), method="euclidean")
b<-dist(t(get(paste("GC",indeks2[i], ".stack", sep=""))), method="euclidean")

a<-mantel(a, b, method = "pearson", permutations = 999)
wynik.mantel[i,1]<-a$statistic
wynik.mantel[i,2]<-a$signif}
wynik.mantel
```

***H3: location invariant measures*. This is verified by observing the distributions of the measures and structural changes between measures, reflected by changes in pair-correlations and matrix correlations. This results in comparison of schemes (3,6), (9,12) and (15,18).**

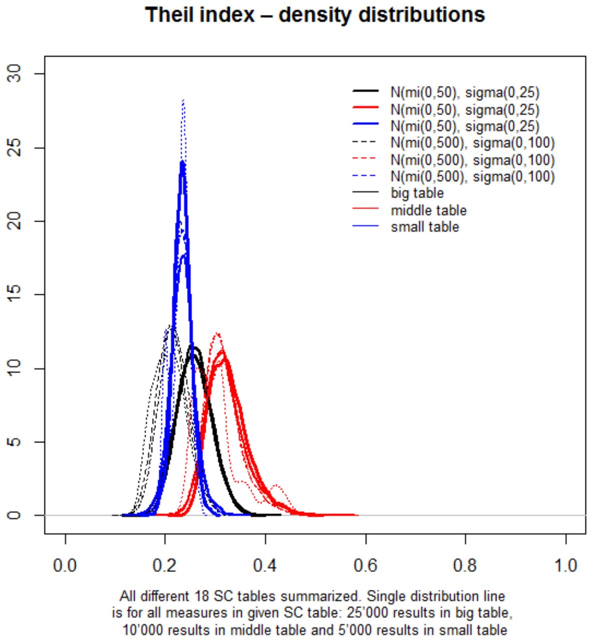
# for corr()

```
wynik.mantel<-matrix(0, nrow=3, ncol=2)
rownames(wynik.mantel)<-c("Big P N1, N2", "Middle P N1 N2","Small P N1 N2")
colnames(wynik.mantel)<-c("Mantel z","p-value")

for(i in 1:3){
indeks1<-c(3,9,15)
indeks2<-c(6,12,18)

a<-cor(get(paste("GC",indeks1[i], ".stack", sep="")))
b<-cor(get(paste("GC",indeks2[i], ".stack", sep="")))

a<-mantel(a, b, method = "pearson", permutations = 999)
wynik.mantel[i,1]<-a$statistic
wynik.mantel[i,2]<-a$signif}
wynik.mantel
```

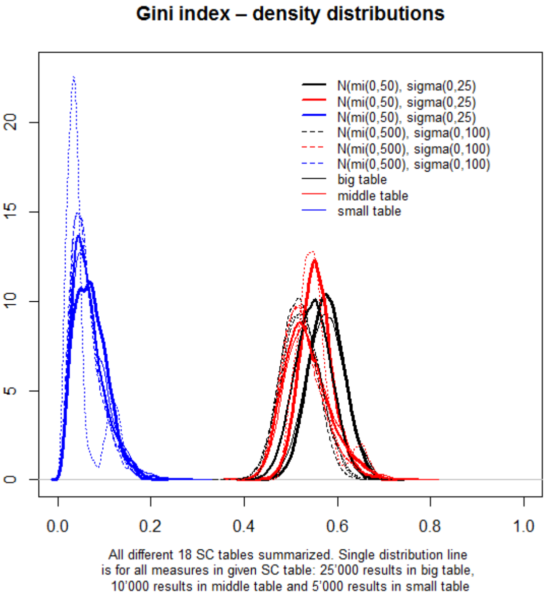
# for dist()

```
wynik.mantel<-matrix(0, nrow=3, ncol=2)
rownames(wynik.mantel)<-c("Big P N1, N2", "Middle P N1 N2","Small P N1 N2")
colnames(wynik.mantel)<-c("Mantel z","p-value")

for(i in 1:3){
indeks1<-c(3,9,15)
indeks2<-c(6,12,18)

a<-dist(t(get(paste("GC",indeks1[i], ".stack", sep=""))), method="euclidean")
b<-dist(t(get(paste("GC",indeks2[i], ".stack", sep=""))), method="euclidean")

a<-mantel(a, b, method = "pearson", permutations = 999)
wynik.mantel[i,1]<-a$statistic
wynik.mantel[i,2]<-a$signif}
wynik.mantel
```

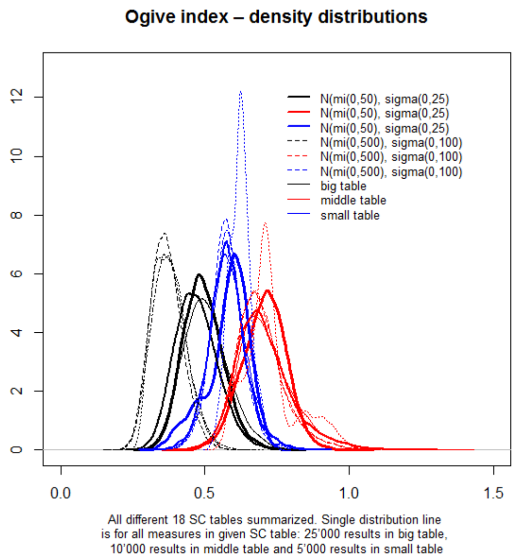
```
# FIG.6A - summary of Theil index - distributions

m=25
range.Theil<-(7*m+1):(8*m)
plot(density(SC1[,range.Theil[1:25]]), xlim=c(0,1), ylim=c(0,30), lty=1, main="",
xlab="", ylab="")
m=10
range.Theil<-(7*m+1):(8*m)
lines(density(SC7[,range.Theil[1:10]]), lty=1, col="red", lwd=1)
m=25
range.Theil<-(7*m+1):(8*m)
lines(density(SC2[,range.Theil[1:25]]), lty=1, lwd=3)
m=10
range.Theil<-(7*m+1):(8*m)
lines(density(SC8[,range.Theil[1:10]]), lty=1, col="red", lwd=3)
m=25
range.Theil<-(7*m+1):(8*m)
lines(density(SC3[,range.Theil[1:25]]), lty=1, lwd=2)
m=10
range.Theil<-(7*m+1):(8*m)
lines(density(SC9[,range.Theil[1:10]]), lty=1, col="red", lwd=2)
m=25
range.Theil<-(7*m+1):(8*m)
lines(density(SC4[,range.Theil[1:25]]), lty=2, lwd=1)
m=10
range.Theil<-(7*m+1):(8*m)
lines(density(SC10[,range.Theil[1:10]]), lty=2, col="red")
m=25
range.Theil<-(7*m+1):(8*m)
lines(density(SC5[,range.Theil[1:25]]), lty=3, lwd=1)
m=10
range.Theil<-(7*m+1):(8*m)
lines(density(SC11[,range.Theil[1:10]]), lty=3, col="red")
m=25
range.Theil<-(7*m+1):(8*m)
lines(density(SC6[,range.Theil[1:25]]), lty=4, lwd=1)
m=10
range.Theil<-(7*m+1):(8*m)
lines(density(SC12[,range.Theil[1:10]]), lty=4, col="red")

m=5
range.Theil<-(7*m+1):(7*m)
lines(density(SC13[,range.Theil[1:5]]), lty=1, col="blue", lwd=1)
m=5
range.Theil<-(7*m+1):(8*m)
lines(density(SC14[,range.Theil[1:5]]), lty=1, col="blue", lwd=3)
m=5
range.Theil<-(7*m+1):(8*m)
lines(density(SC15[,range.Theil[1:5]]), lty=1, col="blue", lwd=2)
m=5
range.Theil<-(7*m+1):(8*m)
lines(density(SC16[,range.Theil[1:5]]), lty=2, col="blue")
m=5
range.Theil<-(7*m+1):(8*m)
lines(density(SC17[,range.Theil[1:5]]), lty=3, col="blue")
m=5
range.Theil<-(7*m+1):(8*m)
lines(density(SC18[,range.Theil[1:5]]), lty=4, col="blue")


legend(0.55, 30, c("N(mi(0,50), sigma(0,25)", "N(mi(0,50), sigma(0,25)", "N(mi(0,50),
sigma(0,25)", "N(mi(0,500), sigma(0,100)", "N(mi(0,500), sigma(0,100)", "N(mi(0,500),
sigma(0,100)", "big table", "middle table", "small table"), lty=c(1,1,1,2,2,2,1,1,1),
lwd=c(2,2,2,1,1,1,1,1,1),
col=c("black","red","blue","black","red","blue","black","red","blue"), cex=0.8,
bty="n")

title(main="Theil index - density distributions", sub="All different 18 SC tables
summarized. Single distribution line
is for all measures in given SC table: 25’000 results in big table,
10’000 results in middle table and 5’000 results in small table", cex.sub=0.8)
```



```
# FIG.6B - summary of Gini index - distributions

m=25
range.Gini<-(12*m+1):(13*m)
plot(density(SC1[,range.Gini[1:25]]), lty=1, xlim=c(0,1), ylim=c(0,23), main="",
xlab="", ylab="")
m=10
range.Gini<-(12*m+1):(13*m)
lines(density(SC7[,range.Gini[1:10]]), lty=1, col="red", lwd=1)
m=5
range.Gini<-(12*m+1):(13*m)
lines(density(SC13[,range.Gini[1:5]]), lty=1, col="blue", lwd=1)

m=25
range.Gini<-(12*m+1):(13*m)
lines(density(SC2[,range.Gini[1:25]]), lty=1, lwd=3)
m=10
range.Gini<-(12*m+1):(13*m)
lines(density(SC8[,range.Gini[1:10]]), lty=1, col="red", lwd=3)
m=5
range.Gini<-(12*m+1):(13*m)
lines(density(SC14[,range.Gini[1:5]]), lty=1, col="blue", lwd=3)

m=25
range.Gini<-(12*m+1):(13*m)
lines(density(SC3[,range.Gini[1:25]]), lty=1, lwd=2)
m=10
range.Gini<-(12*m+1):(13*m)
lines(density(SC9[,range.Gini[1:10]]), lty=1, col="red", lwd=2)
m=5
range.Gini<-(12*m+1):(13*m)
lines(density(SC15[,range.Gini[1:5]]), lty=1, col="blue", lwd=2)

m=25
range.Gini<-(12*m+1):(13*m)
lines(density(SC4[,range.Gini[1:25]]), lty=2, lwd=1)
m=10
range.Gini<-(12*m+1):(13*m)
lines(density(SC10[,range.Gini[1:10]]), lty=2, col="red")
m=5
range.Gini<-(12*m+1):(13*m)
lines(density(SC16[,range.Gini[1:5]]), lty=2, col="blue")

m=25
range.Gini<-(12*m+1):(13*m)
lines(density(SC5[,range.Gini[1:25]]), lty=3, lwd=1)
m=10
range.Gini<-(12*m+1):(13*m)
lines(density(SC11[,range.Gini[1:10]]), lty=3, col="red")
m=5
range.Gini<-(12*m+1):(13*m)
lines(density(SC17[,range.Gini[1:5]]), lty=3, col="blue")

m=25
range.Gini<-(12*m+1):(13*m)
lines(density(SC6[,range.Gini[1:25]]), lty=4, lwd=1)
m=10
range.Gini<-(12*m+1):(13*m)
lines(density(SC12[,range.Gini[1:10]]), lty=4, col="red")
m=5
range.Gini<-(12*m+1):(13*m)
lines(density(SC18[,range.Gini[1:5]]), lty=4, col="blue")

legend(0.5, 23, c("N(mi(0,50), sigma(0,25)", "N(mi(0,50), sigma(0,25)", "N(mi(0,50),
sigma(0,25)", "N(mi(0,500), sigma(0,100)", "N(mi(0,500), sigma(0,100)", "N(mi(0,500),
sigma(0,100)", "big table", "middle table", "small table"), lty=c(1,1,1,2,2,2,1,1,1),
lwd=c(2,2,2,1,1,1,1,1,1),
col=c("black","red","blue","black","red","blue","black","red","blue"), cex=0.8,
bty="n")

title(main="Gini index - density distributions", sub="All different 18 SC tables
summarized. Single distribution line
is for all measures in given SC table: 25’000 results in big table,
10’000 results in middle table and 5’000 results in small table", cex.sub=0.8)
```



```
# FIG.6C - summary of Ogive index - distributions
m=25
range.Ogive<-1:m
plot(density(SC1[,range.Ogive[1:25]]), lty=1, xlim=c(0,1.5), ylim=c(0,13), main="",
xlab="", ylab="")
m=10
range.Ogive<-1:m
lines(density(SC7[,range.Ogive[1:10]]), lty=1, col="red", lwd=1)
m=5
range.Ogive<-1:m
lines(density(SC13[,range.Ogive[1:5]]), lty=1, col="blue", lwd=1)

m=25
range.Ogive<-1:m
lines(density(SC2[,range.Ogive[1:25]]), lty=1, lwd=3)
m=10
range.Ogive<-1:m
lines(density(SC8[,range.Ogive[1:10]]), lty=1, col="red", lwd=3)
m=5
range.Ogive<-1:m
lines(density(SC14[,range.Ogive[1:5]]), lty=1, col="blue", lwd=3)

m=25
range.Ogive<-1:m
lines(density(SC3[,range.Ogive[1:25]]), lty=1, lwd=2)
m=10
range.Ogive<-1:m
lines(density(SC9[,range.Ogive[1:10]]), lty=1, col="red", lwd=2)
m=5
range.Ogive<-1:m
lines(density(SC15[,range.Ogive[1:5]]), lty=1, col="blue", lwd=2)

m=25
range.Ogive<-1:m
lines(density(SC4[,range.Ogive[1:25]]), lty=2, lwd=1)
m=10
range.Ogive<-1:m
lines(density(SC10[,range.Ogive[1:10]]), lty=2, col="red")
m=5
range.Ogive<-1:m
lines(density(SC16[,range.Ogive[1:5]]), lty=2, col="blue")

m=25
range.Ogive<-1:m
lines(density(SC5[,range.Ogive[1:25]]), lty=3, lwd=1)
m=10
range.Ogive<-1:m
lines(density(SC11[,range.Ogive[1:10]]), lty=3, col="red")
m=5
range.Ogive<-1:m
lines(density(SC17[,range.Ogive[1:5]]), lty=3, col="blue")

m=25
range.Ogive<-1:m
lines(density(SC6[,range.Ogive[1:25]]), lty=4, lwd=1)
m=10
range.Ogive<-1:m
lines(density(SC12[,range.Ogive[1:10]]), lty=4, col="red")
m=5
range.Ogive<-1:m
lines(density(SC18[,range.Ogive[1:5]]), lty=4, col="blue")

legend(0.75, 12.5, c("N(mi(0,50), sigma(0,25)", "N(mi(0,50), sigma(0,25)", "N(mi(0,50),
sigma(0,25)", "N(mi(0,500), sigma(0,100)", "N(mi(0,500), sigma(0,100)", "N(mi(0,500),
sigma(0,100)", "big table", "middle table", "small table"), lty=c(1,1,1,2,2,2,1,1,1),
lwd=c(2,2,2,1,1,1,1,1,1),
col=c("black","red","blue","black","red","blue","black","red","blue"), cex=0.8,
bty="n")

title(main="Ogive index – density distributions", sub="All different 18 SC tables
summarized. Single distribution line
is for all measures in given SC table: 25’000 results in big table,
10’000 results in middle table and 5’000 results in small table", cex.sub=0.8)
```



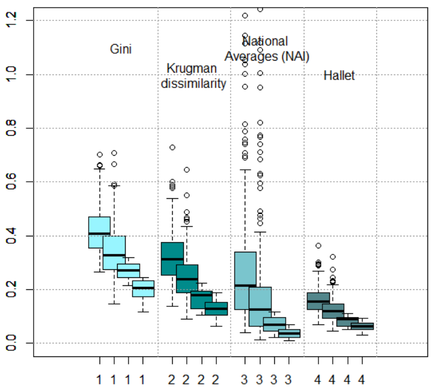
```
############################################
# Empirical Data Analysis

setwd("E:/My simulation")

# m=380 regions, n=86 sectors
lrls<-read.csv("lrls.csv", header=TRUE, sep=";")
rownames(lrls)<-lrls[,1]
lrls<-lrls[,-1]
lrls[lrls==0]<-2
lrls[lrls==1]<-2

# m=380 regions, n=20 sectors
lrss<-read.csv("lrss.csv", header=TRUE, sep=";")
rownames(lrss)<-lrss[,1]
lrss<-lrss[,-1]
lrss[lrss==0]<-2
lrss[lrss==1]<-2

# m=16 regions, n=86 sectors
srls<-read.csv("srls.csv", header=TRUE, sep=";")
rownames(srls)<-srls[,1]
srls<-srls[,-1]
srls[srls==0]<-2
srls[srls==1]<-2

# m=16 regions, n=20 sectors
srss<-read.csv("srss.csv", header=TRUE, sep=";")
rownames(srss)<-srss[,1]
srss<-srss[,-1]
srss[srss==0]<-2
srss[srss==1]<-2

# calculating SC sectoral concentration measures for each region

# for big tables – m=380 regions
m=380
wyniki.lrls<-matrix(0, ncol=13, nrow=m)
a<-c("Ogive", "Krugman dissimilarity", "National Averages (NAI)", "Hallet", "Relative
Diversity (RDI)", "Shannon’s H", "Relative H", "Theil’s H", "Kullback-Leibler
divergence (KLD)", "Refined diversification", "Hachman", "Relative Specialisation (RSI,
max LQ)", "Gini")
colnames(wyniki.lrls)<-a

# for big tables – m=380 regions
m=380
wyniki.lrss<-matrix(0, ncol=13, nrow=m)
a<-c("Ogive", "Krugman dissimilarity", "National Averages (NAI)", "Hallet", "Relative
Diversity (RDI)", "Shannon’s H", "Relative H", "Theil’s H", "Kullback-Leibler
divergence (KLD)", "Refined diversification", "Hachman", "Relative Specialisation (RSI,
max LQ)", "Gini")
colnames(wyniki.lrss)<-a

# for small tables – m=16 regions
m=16
wyniki.srls<-matrix(0, ncol=13, nrow=m)
a<-c("Ogive", "Krugman dissimilarity", "National Averages (NAI)", "Hallet", "Relative
Diversity (RDI)", "Shannon’s H", "Relative H", "Theil’s H", "Kullback-Leibler
divergence (KLD)", "Refined diversification", "Hachman", "Relative Specialisation (RSI,
max LQ)", "Gini")
colnames(wyniki.srls)<-a

m=16
wyniki.srss<-matrix(0, ncol=13, nrow=m)
```

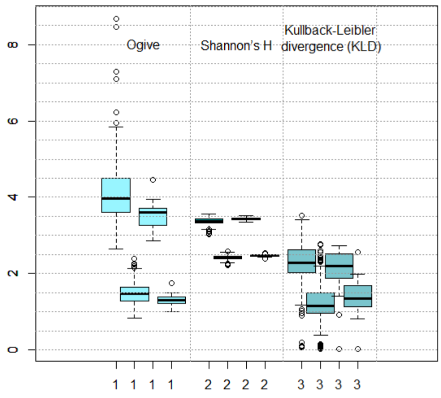
```
a<-c("Ogive", "Krugman dissimilarity", "National Averages (NAI)", "Hallet", "Relative
Diversity (RDI)", "Shannon’s H", "Relative H", "Theil’s H", "Kullback-Leibler
divergence (KLD)", "Refined diversification", "Hachman", "Relative Specialisation (RSI,
max LQ)", "Gini")
colnames(wyniki.srss)<-a

# “FUNCTION” for SC measures
# operates on object „data”,
# requires setting number of rows of matrix (n=number of sectors)
# and number of columns of matrix (m=number of regions)
# and id of iteration (t from 1 to ex.1000)==1
# and the name of final object fin_obj
# pastes the results to prepared SC matrices

SCmeasures<-function(dane, n, m, t, fin_obj) {

#n<-86 # number of sectors
#m<-380 # number of regions
mm<-1 # constant for ranges
#fin_obj<-wyniki.all

#dane<-matrix(data, nrow=n, ncol=m,  byrow = FALSE)
#rownames(dane)<-paste(rep("sector", times=n),1:n)
#colnames(dane)<-paste(rep("region", times=m), 1:m)
#dane<-lrls
sum.by.regions<-colSums(dane)
dane<-rbind(dane, sum.by.regions)
sum.by.sectors<-rowSums(dane)
dane<-cbind(dane, sum.by.sectors)
dane

empl<-dane[1:n, 1:m]
sectors<-rownames(dane[1:n,])

# data ranges for output
range.Ogive<-1:mm
range.Krugman<-(mm+1):(2*mm)
range.NAI<-(2*mm+1):(3*mm)
range.Hallet<-(3*mm+1):(4*mm)
range.RDI<-(4*mm+1):(5*mm)
range.Shannon<-(5*mm+1):(6*mm)
range.RelativeH<-(6*mm+1):(7*mm)
range.Theil<-(7*mm+1):(8*mm)
range.KLD<-(8*mm+1):(9*mm)
range.divers<-(9*mm+1):(10*mm)
range.Hachman<-(10*mm+1):(11*mm)
range.RSI<-(11*mm+1):(12*mm)
range.Gini<-(12*mm+1):(13*mm)

#################################\
# common elements of calculations
# matrix of industrial shares by regions
shares.in.columns<-matrix(0, ncol=m, nrow=n)
for(i in 1:n){
temp<-empl[i,]/sum.by.regions[1:m]
shares.in.columns[i,]<-as.matrix(temp)}
colSums(shares.in.columns) # check of sums in columns

# vector of industrial shares for national economy
share.in.columns.extra<-sum.by.sectors/sum.by.sectors[n+1]

#################################\
# Ogive index
wynik<-matrix(0, ncol=m, nrow=1)
colnames(wynik)<-paste("region",1:m)
```

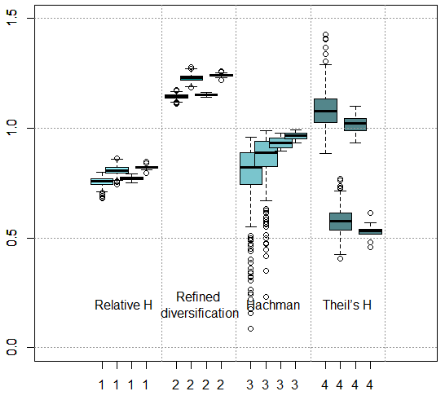
```
rownames(wynik)<-c("Ogive index")

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
fin_obj[1:m,range.Ogive]<-wynik
################################\

# Krugman dissimilarity index, NAI, Hallet, Relative Diversity Index
wynik<-matrix(0, ncol=m, nrow=4)
colnames(wynik)<-paste("region",1:m)
rownames(wynik)<-c("Krugman Dissimilarity", "National Averages (NAI)", "Hallet",
"Relative Diversity (RDI)")
for(i in 1:m){
diff.abs<-abs(shares.in.columns[,i]- share.in.columns.extra[1:n])
wynik[1,i]<-sum(diff.abs)}

for(i in 1:m){
diff.abs<-(shares.in.columns[,i]- share.in.columns.extra[1:n])^2/
share.in.columns.extra[1:n]
wynik[2,i]<-sum(diff.abs)}
wynik[3,]<-wynik[1,]/2
wynik[4,]<-1/wynik[1,]
wynik
fin_obj[1:m,range.Krugman]<-wynik[1,]
fin_obj[1:m,range.NAI]<-wynik[2,]
fin_obj[1:m,range.Hallet]<-wynik[3,]
fin_obj[1:m,range.RDI]<-wynik[4,]

################################\
# 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")

logs.matrix<-matrix(0, ncol=m, nrow=n)
logs.matrix<-shares.in.columns*log(shares.in.columns)
head(logs.matrix)
sum.columns<-colSums(logs.matrix)
head(sum.columns)

logs.matrix.kld<-matrix(0, ncol=m, nrow=n)
logs.matrix.kld<-shares.in.columns* log(shares.in.columns/ share.in.columns.extra)
sum.columns.kld<-colSums(logs.matrix.kld)
equal.dist.H<-abs((1/n)*log(1/n)*n)

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)
head(wynik)
fin_obj[1:m,range.Shannon]<-wynik[1,]
fin_obj[1:m,range.RelativeH]<-wynik[2,]
fin_obj[1:m,range.Theil]<-wynik[3,]
fin_obj[1:m,range.KLD]<-wynik[4,]

################################\
# diversification index
wynik<-matrix(0, ncol=m, nrow=1)
colnames(wynik)<-paste("region",1:m)
rownames(wynik)<-c("Refined Diversification")
```

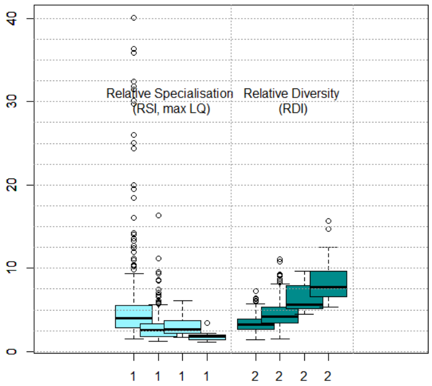
```
hyp.con<-c(100, rep(0, times=n-1))

for(j in 1:m){
vec1<- shares.in.columns[,j]*100
vec2<- share.in.columns.extra*100
vec1.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)
index<-(vec1.sum- vec2.sum)/( vec3.sum- vec2.sum)
wynik[1,j]<-index}
wynik
fin_obj[1:m,range.divers]<-wynik

################################\
# LQ, Hachman, RSI
wynik<-matrix(0, ncol=m, nrow=2)
colnames(wynik)<-paste("region",1:m)
rownames(wynik)<-c("Hachman", "Relative Specialisation (RSI, max LQ)")

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[1:n]
LQ[,j]<-vec1/vec2}

temp<-LQ*shares.in.columns
temp1<-colSums(temp)
wynik[1,]<-1/temp1
wynik[2,]<-apply(LQ, 2, max) # max po kolumnach
wynik
fin_obj[1:m,range.Hachman]<-wynik[1,]
fin_obj[1:m,range.RSI]<-wynik[2,]

################################\
# Gini
wynik<-matrix(0, ncol=m, nrow=1)
colnames(wynik)<-paste("region",1:m)
rownames(wynik)<-c("Gini")

for(j in 1:m){
vec1<- shares.in.columns[,j]
vec2<- share.in.columns.extra
vec3<-vec1/vec2
vec3.av<-mean(vec3)
vec3.sort<-sort(vec3, decreasing=TRUE)
vec4<-1:n
vec5<-abs(vec3.sort-vec3.av)*vec4
tot.vec5<-sum(vec5)
wynik[1,j]<-(tot.vec5*2)/(n*n* vec3.av)}
wynik
fin_obj[1:m,range.Gini]<-wynik
return(fin_obj)[1:m,]

} # closing the loop for the function

################################\
# calling SCmeasures() function
# “(SCmeasures<-function(dane, n, m, t, fin_obj))”

fin_obj<-wyniki.lrls
wyniki.lrls<-SCmeasures(lrls, 86, 380, 1, fin_obj=wyniki.lrls)
summary(wyniki.lrls)
a<-colSums(lrls)
aa<-sum(a)
aaa<-a/aa
plot(aaa, wyniki.lrls[,13]) # zależność Gini od wielkości regionu

###############
fin_obj<-wyniki.lrss
wyniki.lrss<-SCmeasures(lrss, 20, 380, 1, fin_obj=wyniki.lrss)
summary(wyniki.lrss)

a<-colSums(lrss)
aa<-sum(a)
aaa<-a/aa
plot(aaa, wyniki.lrss[,13]) # zależność Gini od wielkości regionu

###################
fin_obj<-wyniki.srls
wyniki.srls<-SCmeasures(srls, 86, 16, 1, fin_obj=wyniki.srls)
summary(wyniki.srls)

a<-colSums(srls)
aa<-sum(a)
aaa<-a/aa
plot(aaa, wyniki.srls[,13]) # zależność Gini od wielkości regionu

#####################
fin_obj<-wyniki.srss
wyniki.srss<-SCmeasures(srss, 20, 16, 1, fin_obj=wyniki.srss)
summary(wyniki.srss)

a<-colSums(srss)
aa<-sum(a)
aaa<-a/aa
plot(aaa, wyniki.srss[,13]) # zależność Gini od wielkości regionu

################


boxplot(wyniki.lrls[,c(13,2,3,4)], at=(1:4)-0.3, boxwex=0.3, names=1:4, ylim=c(0,1.2),
xlim=c(0,5), col=c("cadetblue1","darkcyan","cadetblue3","cadetblue4"))
boxplot(wyniki.lrss[,c(13,2,3,4)], at=(1:4)-0.1, add=TRUE, boxwex=0.3, names=1:4,
col=c("cadetblue1","darkcyan","cadetblue3","cadetblue4"))
boxplot(wyniki.srls[,c(13,2,3,4)], at=(1:4)+0.1, add=TRUE, boxwex=0.3, names=1:4,
col=c("cadetblue1","darkcyan","cadetblue3","cadetblue4"))
boxplot(wyniki.srss[,c(13,2,3,4)], at=(1:4)+0.3, add=TRUE, boxwex=0.3, names=1:4,
col=c("cadetblue1","darkcyan","cadetblue3","cadetblue4"))
abline(h=c(0:10)*0.2, lty=3, col="grey60")
abline(v=c(1:5)+0.5, lty=3, col="grey60")
text(1,1.1, c("Gini"))
text(2,1.0, c("Krugman
dissimilarity"))
text(3,1.1, c("National
Averages (NAI)"))
text(4,1.0, c("Hallet"))

boxplot(wyniki.lrls[,c(1,6,9)], at=(1:3)-0.3, boxwex=0.3, names=1:3, xlim=c(0,4),
col=c("cadetblue1","darkcyan","cadetblue3","cadetblue4"))
```

```
boxplot(wyniki.lrss[,c(1,6,9)], at=(1:3)-0.1, add=TRUE, boxwex=0.3, names=1:3,
col=c("cadetblue1","darkcyan","cadetblue3","cadetblue4"))
boxplot(wyniki.srls[,c(1,6,9)], at=(1:3)+0.1, add=TRUE, boxwex=0.3, names=1:3,
col=c("cadetblue1","darkcyan","cadetblue3","cadetblue4"))
boxplot(wyniki.srss[,c(1,6,9)], at=(1:3)+0.3, add=TRUE, boxwex=0.3, names=1:3,
col=c("cadetblue1","darkcyan","cadetblue3","cadetblue4"))
abline(h=c(0:40)*0.5, lty=3, col="grey60")
abline(v=c(1:5)+0.5, lty=3, col="grey60")
text(1,8.0, c("Ogive"))
text(2,8.0, c("Shannon’s H"))
text(3,8.2, c("Kullback-Leibler
 divergence (KLD)"))

boxplot(wyniki.lrls[,c(7,10,11,8)], at=(1:4)-0.3, boxwex=0.3, names=1:4,
xlim=c(0,5),ylim=c(0,1.5), col=c("cadetblue1","darkcyan","cadetblue3","cadetblue4"))
boxplot(wyniki.lrss[,c(7,10,11,8)], at=(1:4)-0.1, add=TRUE, boxwex=0.3, names=1:4,
col=c("cadetblue1","darkcyan","cadetblue3","cadetblue4"))
boxplot(wyniki.srls[,c(7,10,11,8)], at=(1:4)+0.1, add=TRUE, boxwex=0.3, names=1:4,
col=c("cadetblue1","darkcyan","cadetblue3","cadetblue4"))
boxplot(wyniki.srss[,c(7,10,11,8)], at=(1:4)+0.3, add=TRUE, boxwex=0.3, names=1:4,
col=c("cadetblue1","darkcyan","cadetblue3","cadetblue4"))
abline(h=c(0:40)*0.5, lty=3, col="grey60")
abline(v=c(1:5)+0.5, lty=3, col="grey60")
text(1,0.2, c("Relative H"))
text(2,0.2, c("Refined
 diversification"))
text(3,0.2, c("Hachman"))
text(4,0.2, c("Theil’s H"))


boxplot(wyniki.lrls[,c(12,5)], at=(1:2)-0.3, boxwex=0.3, names=1:2, xlim=c(0,3),
col=c("cadetblue1","darkcyan","cadetblue3","cadetblue4"))
boxplot(wyniki.lrss[,c(12,5)], at=(1:2)-0.1, add=TRUE, boxwex=0.3, names=1:2,
col=c("cadetblue1","darkcyan","cadetblue3","cadetblue4"))
boxplot(wyniki.srls[,c(12,5)], at=(1:2)+0.1, add=TRUE, boxwex=0.3, names=1:2,
col=c("cadetblue1","darkcyan","cadetblue3","cadetblue4"))
boxplot(wyniki.srss[,c(12,5)], at=(1:2)+0.3, add=TRUE, boxwex=0.3, names=1:2,
col=c("cadetblue1","darkcyan","cadetblue3","cadetblue4"))
text(1,30.2, c("Relative Specialisation
 (RSI, max LQ)"))
text(2,30.2, c("Relative Diversity
 (RDI)"))
abline(h=c(0:40)*2.5, lty=3, col="grey60")
abline(v=c(1:5)+0.5, lty=3, col="grey60")
```



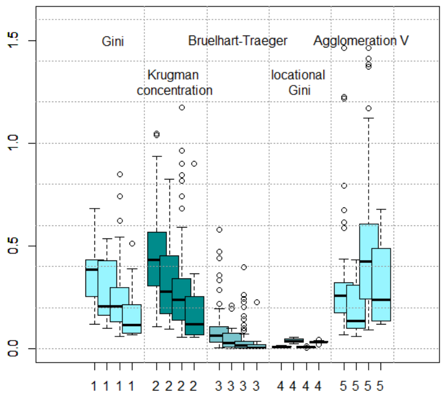



0.0
0.5
1.0
1.5
Relative H
Refined diversification
Hachman
Theil's H
1 1 1 1
2 2 2 2
3 3 3 3
4 4 4 4

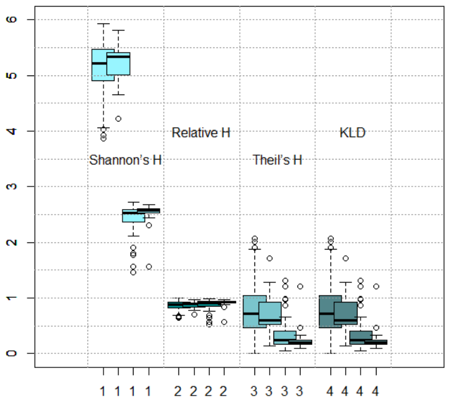

0
10
20
30
40
Relative Specialisation (RSI, max LQ)
Relative Diversity (RDI)
1 1 1 1
2 2 2 2

```
# calculating GC sectoral concentration measures
# for each matrix generated above and collecting the measures in matrices
# generating empty matrices for measures for each scheme (in total 18 matrices of GC
measures)
# GC1 is for scheme 1

# for big tables - n=86 industries
n=86
wyniki.lrls<-matrix(0, ncol=9, nrow=n)
a<-c("Gini", "Krugman Concentration", "Bruelhart-Traeger", "locational Gini",
"Agglomeration V", "Shannon’s H", "Relative H", "Theil’s H", "KLD")
colnames(wyniki.lrls)<-a

n=86
wyniki.srls<-matrix(0, ncol=9, nrow=n)
a<-c("Gini", "Krugman Concentration", "Bruelhart-Traeger", "locational Gini",
"Agglomeration V", "Shannon’s H", "Relative H", "Theil’s H", "KLD")
colnames(wyniki.srls)<-a

# for small tables - n=20 industries

n=20
wyniki.lrss<-matrix(0, ncol=9, nrow=n)
a<-c("Gini", "Krugman Concentration", "Bruelhart-Traeger", "locational Gini",
"Agglomeration V", "Shannon’s H", "Relative H", "Theil’s H", "KLD")
colnames(wyniki.lrss)<-a

n=20
wyniki.srss<-matrix(0, ncol=9, nrow=n)
a<-c("Gini", "Krugman Concentration", "Bruelhart-Traeger", "locational Gini",
"Agglomeration V", "Shannon’s H", "Relative H", "Theil’s H", "KLD")
colnames(wyniki.srss)<-a


# “FUNCTION” for GC measures

# requires setting number of rows of matrix (n=number of sectors)
# and number of columns (number of measures)
# and the name of final object fin_obj

GCmeasures<-function(dane, n, m, t, fin_obj) {

mm<-1
sum.by.regions<-colSums(dane)
dane<-rbind(dane, sum.by.regions)
sum.by.sectors<-rowSums(dane)
dane<-cbind(dane, sum.by.sectors)
dane

empl<-dane[1:n, 1:m]
sectors<-rownames(dane[1:n,])

#################################\
# data ranges for output
range.GiniGC<-1:mm
range.KrugmanGC<-(mm+1):(2*mm)
range.BT<-(2*mm+1):(3*mm)
range.locGini<-(3*mm+1):(4*mm)
range.AgglomerationV<-(4*mm+1):(5*mm)
range.ShannonGC<-(5*mm+1):(6*mm)
range.RelativeHGC<-(6*mm+1):(7*mm)
range.TheilGC<-(7*mm+1):(8*mm)
range.KLDGC<-(8*mm+1):(9*mm)
#################################\
#################################\
# common elements of calculations
# matrix of industrial shares by regions
shares.in.rows<-matrix(0, ncol=m, nrow=n)
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 rows

# vector of industrial shares for national economy
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)}
shares.in.columns
colSums(shares.in.columns) # check of sums in columns

share.in.columns.extra<-sum.by.sectors/sum.by.sectors[n+1]
share.in.columns.extra

#################################\
# Gini
wynik<-matrix(0, ncol=n, nrow=1)
colnames(wynik)<-paste("sector",1:n)
rownames(wynik)<-c("Gini")
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<-mean(vec3, na.rm=FALSE)
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
fin_obj[1:n,range.GiniGC]<-wynik


#################################\
# Krugman concentration index
wynik<-matrix(0, ncol=n, nrow=1)
colnames(wynik)<-paste("sector",1:n)
rownames(wynik)<-c("Krugman Concentration")

for(i in 1:n){
diff.abs<-abs(shares.in.rows[i,]-share.in.rows.extra)
wynik[1,i]<-sum(diff.abs)}
wynik
fin_obj[1:n,range.KrugmanGC]<-wynik

#################################\
# Bruelhart -Traeger index
wynik<-matrix(0, ncol=n, nrow=1)
colnames(wynik)<-paste("sector",1:n)
rownames(wynik)<-c("Bruelhart-Traeger")

temp<-matrix(0, nrow=n, ncol=m)
for(i in 1:n){
temp[i,]<-as.matrix(shares.in.rows[i,]/share.in.rows.extra)}

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)}
temp2

wynik<-rowSums(temp2)
wynik
fin_obj[1:n,range.BT]<-wynik

################################\
# locational Gini
wynik<-matrix(0, ncol=n, nrow=1)
colnames(wynik)<-paste("sector",1:n)
rownames(wynik)<-c("locGini")

temp<-matrix(0, nrow=n, ncol=m)
for(i in 1:n){
temp[i,]<-as.matrix(shares.in.columns[i,]/share.in.rows.extra)}
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
fin_obj[1:n,range.locGini]<-wynik

################################\
# Agglomeration V
wynik<-matrix(0, ncol=n, nrow=1)
colnames(wynik)<-paste("sector",1:n)
rownames(wynik)<-c("Agglomeration V")

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-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
fin_obj[1:n,range.AgglomerationV]<-wynik

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

# 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")

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

fin_obj[1:n,range.ShannonGC]<-wynik[1,]
fin_obj[1:n,range.RelativeHGC]<-wynik[2,]
fin_obj[1:n,range.TheilGC]<-wynik[3,]
fin_obj[1:n,range.KLDGC]<-wynik[4,]

######
return(fin_obj)[1:n,]
} # closing the loop for the function
################################\


# calling GCmeasures() function
# “(GCmeasures<-function(dane, n, m, t, fin_obj))”
# big tables
fin_obj<-wyniki.lrls
wyniki.lrls<-GCmeasures(lrls, 86, 380, 1, fin_obj=wyniki.lrls)
summary(wyniki.lrls)

fin_obj<-wyniki.srls
wyniki.srls<-GCmeasures(srls, 86, 16, 1, fin_obj=wyniki.srls)
summary(wyniki.srls)

fin_obj<-wyniki.lrss
wyniki.lrss<-GCmeasures(lrss, 20, 380, 1, fin_obj=wyniki.lrss)
summary(wyniki.lrss)

fin_obj<-wyniki.srss
wyniki.srss<-GCmeasures(srss, 20, 16, 1, fin_obj=wyniki.srss)
summary(wyniki.srss)

boxplot(wyniki.lrls[,c(1,2,3,4,5)], at=(1:5)-0.3, boxwex=0.3, names=1:5, ylim=c(0,1.6),
xlim=c(0,6), col=c("cadetblue1","darkcyan","cadetblue3","cadetblue4"))
boxplot(wyniki.lrss[,c(1,2,3,4,5)], at=(1:5)-0.1, add=TRUE, boxwex=0.3, names=1:5,
col=c("cadetblue1","darkcyan","cadetblue3","cadetblue4"))
boxplot(wyniki.srls[,c(1,2,3,4,5)], at=(1:5)+0.1, add=TRUE, boxwex=0.3, names=1:5,
col=c("cadetblue1","darkcyan","cadetblue3","cadetblue4"))
boxplot(wyniki.srss[,c(1,2,3,4,5)], at=(1:5)+0.3, add=TRUE, boxwex=0.3, names=1:5,
col=c("cadetblue1","darkcyan","cadetblue3","cadetblue4"))
abline(h=c(0:10)*0.2, lty=3, col="grey60")
abline(v=c(1:5)+0.5, lty=3, col="grey60")
text(1,1.5, c("Gini"))
text(2,1.3, c("Krugman
concentration"))
text(3,1.5, c("Bruelhart-Traeger"))
text(4,1.3, c("locational
Gini"))
text(5,1.5, c("Agglomeration V"))


boxplot(wyniki.lrls[,c(6,7,8,9)], at=(1:4)-0.3, boxwex=0.3, names=1:4, ylim=c(0,6),
xlim=c(0,5), col=c("cadetblue1","darkcyan","cadetblue3","cadetblue4"))
boxplot(wyniki.lrss[,c(6,7,8,9)], at=(1:4)-0.1, add=TRUE, boxwex=0.3, names=1:4,
col=c("cadetblue1","darkcyan","cadetblue3","cadetblue4"))
boxplot(wyniki.srls[,c(6,7,8,9)], at=(1:4)+0.1, add=TRUE, boxwex=0.3, names=1:4,
col=c("cadetblue1","darkcyan","cadetblue3","cadetblue4"))
boxplot(wyniki.srss[,c(6,7,8,9)], at=(1:4)+0.3, add=TRUE, boxwex=0.3, names=1:4,
col=c("cadetblue1","darkcyan","cadetblue3","cadetblue4"))
abline(h=c(0:12)*0.5, lty=3, col="grey60")
abline(v=c(0:5)+0.5, lty=3, col="grey60")
text(1,3.5, c("Shannon’s H"))
text(2,4.0, c("Relative H"))
text(3,3.5, c("Theil’s H"))
text(4,4.0, c("KLD"))
```



0.0
0.5
1.0
1.5
Gini
Krugman concentration
Bruelhart-Traeger
locational Gini
Agglomeration V
1 1 1 1
2 2 2 2
3 3 3 3
4 4 4 4
5 5 5 5



0
1
2
3
4
5
6
Shannon's H
Relative H
Theil's H
KLD
1 1 1 1
2 2 2 2
3 3 3 3
4 4 4 4