R codes to paper

**Entropy of tessellated point-pattern**

**as a measure of agglomeration**

by Katarzyna Kopczewska (kkopczewska@wne.uw.edu.pl)

University of Warsaw, Faculty of Economic Sciences

**Codes and data at** [**https://github.com/kkopczewska/entropy**](https://github.com/kkopczewska/entropy)

Spis treści

[# Starter 2](#_Toc58529423)

[# Fig.1 Maximum entropy depending on sample size 3](#_Toc58529424)

[# Fig.2 Tesselations for different point-patterns 4](#_Toc58529425)

[# tesselation for original point pattern – centroids of NTS4 territorial units 4](#_Toc58529426)

[# Fig. 2a tesselation for regular points (as centroids of grid) 4](#_Toc58529427)

[# Fig. 2b tesselation for random points 5](#_Toc58529428)

[# Fig. 2c tesselation for random points agglomerated centrally 6](#_Toc58529429)

[# Fig. 2d tesselation for random points agglomerated at the border 6](#_Toc58529430)

[# Monte Carlo simulation of relH for different locations of agglomerated points 7](#_Toc58529431)

[# Fig.4 Monte Carlo simulation of relH for different locations of agglomerated points 10](#_Toc58529432)

[# Tab.1 – repeating the excerisize by Lews(1979) - validation 12](#_Toc58529433)

[# Lews (1979) results 12](#_Toc58529434)

[# Tab.2 – simulation of relH and Theil 14](#_Toc58529435)

[# CRS simulation, in PL contour 14](#_Toc58529436)

[# empirical data simulation – business locations in Warsaw 14](#_Toc58529437)

[# extra: theoretical simulation – random point within box 15](#_Toc58529438)

[# Business locations in Warsaw city 17](#_Toc58529439)

[# Fig.6a - Entropy for tesselation of empirical data 17](#_Toc58529440)

[# Fig.3 – Normality of Shannon’s H distribution 17](#_Toc58529441)

[# extra: simulation – relative entropy for various sample size n 19](#_Toc58529442)

[# Fig.6b - Analysis of gridded population in Warsaw city 20](#_Toc58529443)

[# Fig.5b – Business location in Lubelskie region 23](#_Toc58529444)

[# Fig.5a –Business locations in Silesian region 25](#_Toc58529445)

[# Fig.6c – Business location in Silesian conurbation (city) 27](#_Toc58529446)

[# Fig.6d - Analysis of gridded population in Silesian metropolitan area 31](#_Toc58529447)

[# Fig.6e – Business locations in Lublin city 33](#_Toc58529448)

[# Fig.6f - Analysis of gridded population in Lublin city 34](#_Toc58529449)

# # Starter

**# packages**

library(sp)

library(spdep)

library(spatialreg)

library(rgdal)

library(maptools)

library(spatstat)

**# reading the data – shapefile maps**

setwd("D:/data") # set working directory on your computer

# reading countour maps for Poland

pl<-readOGR(".", "Panstwo") # map PL NTS0

woj<-readOGR(".", "wojewodztwa") # map PL NTS2

pow<-readOGR(".", "powiaty") # map PL NTS4

crds<-coordinates(pow)

# reading shapefile for Warsaw

waw<-readOGR(".", "waw")

waw<-spTransform(waw, CRS("+proj=longlat +datum=NAD83"))

# reading shapefile grid for population

pop<-readOGR(".", "PD\_STAT\_GRID\_CELL\_2011") # takes long!

pop<-spTransform(pop, CRS("+proj=longlat +datum=NAD83"))

pop.df<-as.data.frame(pop) # wyodrębnienie danych

pop.grid<-as(pop, "SpatialPolygons") # wyłącznie grid

# conversion to numeric

for(i in 1:12){

pop.df[,i]<-as.numeric(as.character(pop.df[,i]))}

**# reading point data on business location for cities**

# reading and cleaning dataset for Warsaw

ROA<-read.table("ORBIS.csv", sep = ";", header=TRUE, dec=",")

bb<-bbox(waw)

b2<-which(is.na(ROA[,4])==TRUE)

b3<-which(ROA[,4]<=bb[1,1] | ROA[,4]>=bb[1,2])

b4<-which(ROA[,3]<=bb[2,1] | ROA[,3]>=bb[2,2])

ROA<-ROA[c(-b2, -b3, -b4),]

ROA$ones<-rep(1, times=dim(ROA)[1])

ROA$lat<-ROA$lat+rnorm(dim(ROA)[1], mean=0, sd=0.01)

ROA$lon<-ROA$lon+rnorm(dim(ROA)[1], mean=0, sd=0.01)

# reading dataset for Lubelskie region

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

# reading dataset for Silesia region

silesia<-read.csv("dane\_slaskie\_FULL.csv", header=TRUE, dec=".", sep=",")

# # Fig.1 Maximum entropy depending on sample size

**# entropy depending on number of units**

vec<-c(1,10,100,1000,10000,100000,1000000,10000000)

entropy<-matrix(0, nrow=8, ncol=1)

for(i in 1:8){

n<-vec[i]

ent.ref<-log(1/n)\*(-1)

ent.ref

entropy[i,1]<-ent.ref

}

plot(entropy, type="l", xlim=c(0,9))

points(entropy, pch=21, bg="black")

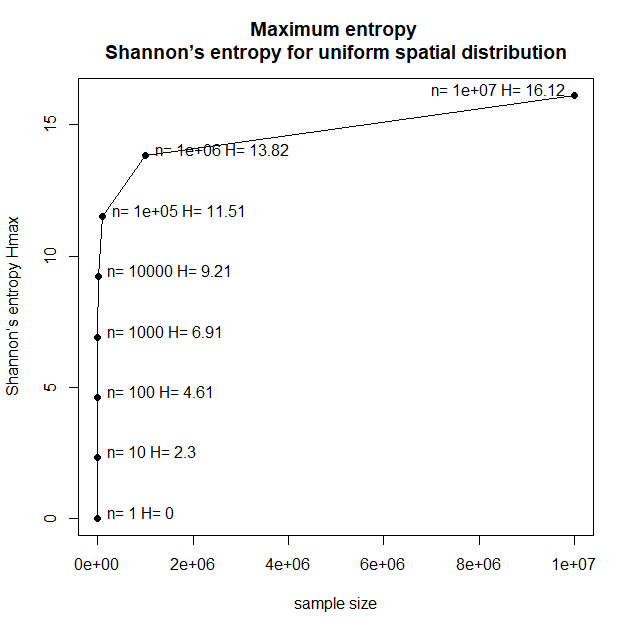
text((1:8)+0.01, entropy+0.2, labels=as.character(as.numeric(vec)))

plot(vec, entropy, type="l", main="Maximum entropy

Shannon’s entropy for uniform spatial distribution", xlab="sample size", ylab="Shannon’s entropy Hmax")

points(vec, entropy, pch=21, bg="black")

text(vec+10, entropy+0.2, labels=paste("n=",vec, "H=",round(entropy,2)), pos=c(4,4,4,4,4,4,4,2))



# # Fig.2 Tesselations for different point-patterns

## # tesselation for original point pattern – centroids of NTS4 territorial units

pl.owin<-as(pl, "owin")

pl.ppp<-ppp(x=crds[,1], y=crds[,2], window=pl.owin)

pl.tes<-dirichlet(pl.ppp) # Dirichlet tessellation

a<-tile.areas(pl.tes)

a1<-a/sum(a)

ent1<-sum(-1\*a1\*log(a1)) # Shannon

n<-length(a)

ent.max<-log(1/n)\*(-1)

ent.max

ent.max2<-log(n) # the same as ent.max

ent.max2

ent.rel<-ent1/ent.max # Relative H

ent.rel

ent.rel2<-ent.max-ent1 # Theil redundancy

ent.rel2

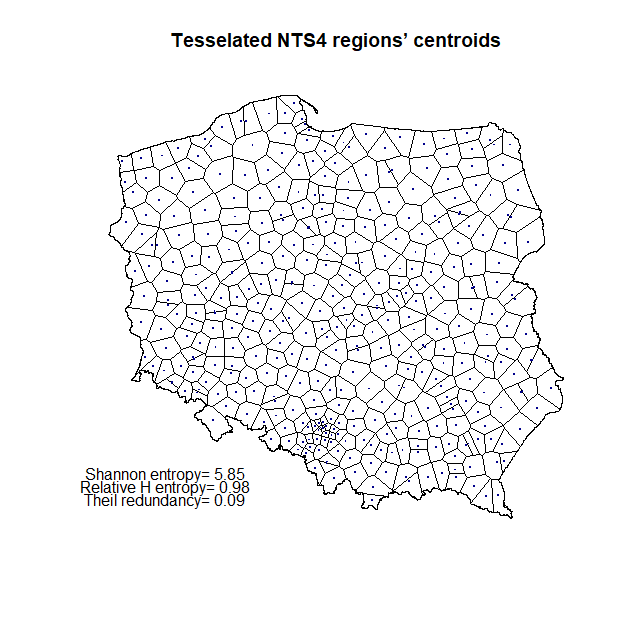
plot(pl.tes, main="Tesselated NTS4 regions’ centroids")

plot(pl.ppp, add=TRUE, pch=".", col="darkblue", cex=2)

text(257000, 200000, paste("Shannon entropy=", round(ent1,2)))

text(257000, 180000, paste("Relative H entropy=", round(ent.rel,2)))

text(257000, 160000, paste("Theil redundancy=", round(ent.rel2,2)))



## # Fig. 2a tesselation for regular points (as centroids of grid)

b1<-spsample(pl, 380, "regular")

pl.owin<-as(pl, "owin")

pl.ppp<-ppp(x=b1$x1, y=b1$x2, window=pl.owin)

pl.tes<-dirichlet(pl.ppp) # Dirichlet tessellation

a<-tile.areas(pl.tes)

a1<-a/sum(a)

ent1<-sum(-1\*a1\*log(a1)) # Shannon

n<-length(a)

ent.max<-log(1/n)\*(-1)

ent.max

ent.max2<-log(n) # the same as ent.max

ent.max2

ent.rel<-ent1/ent.max # Relative H

ent.rel

ent.rel2<-ent.max-ent1 # Theil redundancy

ent.rel2

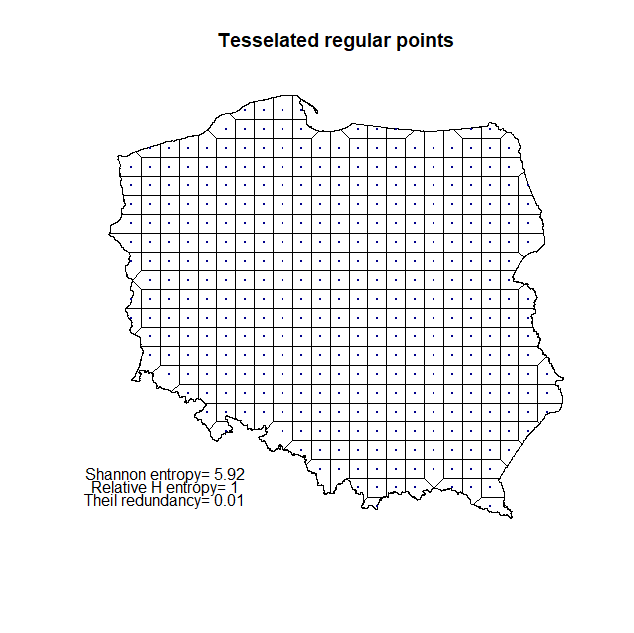
plot(pl.tes, main="Tesselated regular points")

plot(pl.ppp, add=TRUE, pch=".", col="darkblue", cex=2)

text(257000, 200000, paste("Shannon entropy=", round(ent1,2)))

text(257000, 180000, paste("Relative H entropy=", round(ent.rel,2)))

text(257000, 160000, paste("Theil redundancy=", round(ent.rel2,2)))



## # Fig. 2b tesselation for random points

b1<-spsample(pl, 380, "random")

pl.owin<-as(pl, "owin")

pl.ppp<-ppp(x=b1@coords[,1], y=b1@coords[,2], window=pl.owin)

pl.tes<-dirichlet(pl.ppp) # Dirichlet tessellation

a<-tile.areas(pl.tes)

a1<-a/sum(a)

ent1<-sum(-1\*a1\*log(a1)) # Shannon

n<-length(a)

ent.max<-log(1/n)\*(-1)

ent.max

ent.rel<-ent1/ent.max # Relative H

ent.rel

ent.rel2<-ent.max-ent1 # Theil redundancy

ent.rel2

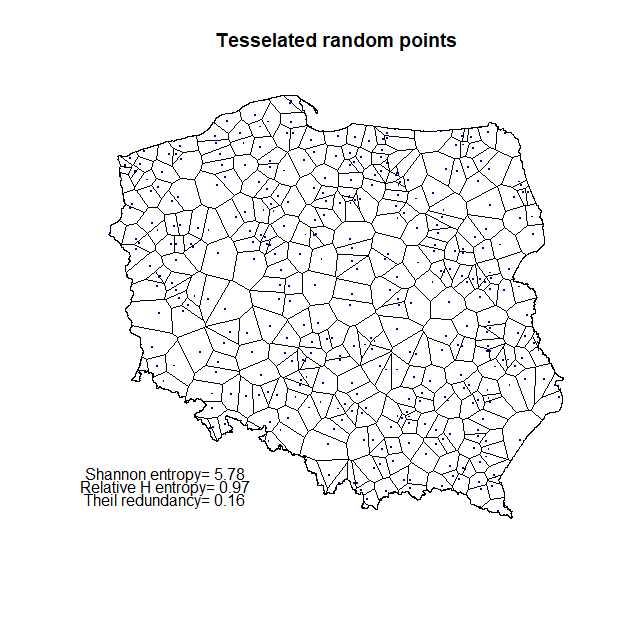
plot(pl.tes, main="Tesselated random points")

plot(pl.ppp, add=TRUE, pch=".", col="darkblue", cex=2)

text(257000, 200000, paste("Shannon entropy=", round(ent1,2)))

text(257000, 180000, paste("Relative H entropy=", round(ent.rel,2)))

text(257000, 160000, paste("Theil redundancy=", round(ent.rel2,2)))



## # Fig. 2c tesselation for random points agglomerated centrally

x1<-rnorm(380, 500000, 15)

y1<-rnorm(380, 460000, 15)

pl.owin<-as(pl, "owin")

pl.ppp<-ppp(x=x1, y=y1, window=pl.owin)

plot(pl.ppp, pch=".")

pl.tes<-dirichlet(pl.ppp) # Dirichlet tessellation

a<-tile.areas(pl.tes)

a1<-a/sum(a)

ent1<-sum(-1\*a1\*log(a1))

n<-length(a)

ent.ref<-log(1/n)\*(-1)

ent.ref

ent.rel<-ent1/ent.ref # Relative H

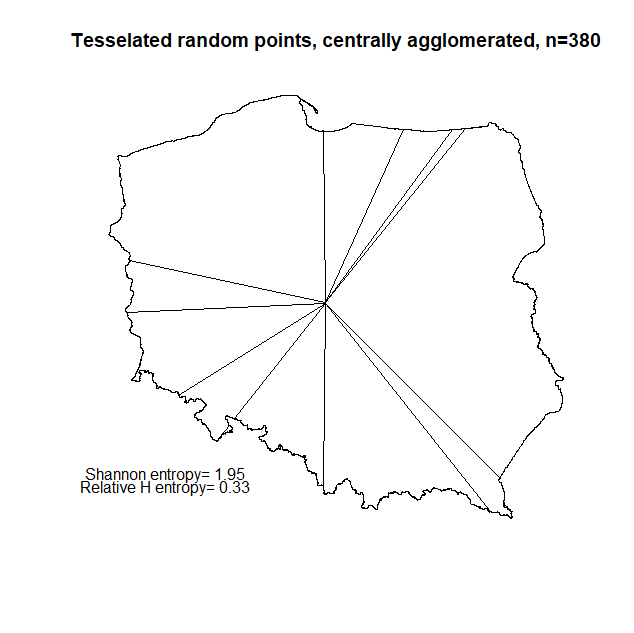
ent.rel

plot(pl.tes, main="Tesselated random points, centrally agglomerated, n=380")

plot(pl.ppp, add=TRUE, pch=".", col="darkblue", cex=2)

text(257000, 200000, paste("Shannon entropy=", round(ent1,2)))

text(257000, 180000, paste("Relative H entropy=", round(ent.rel,2)))



## # Fig. 2d tesselation for random points agglomerated at the border

x1<-rnorm(380, 755000, 10)

y1<-rnorm(380, 160000, 10)

pl.owin<-as(pl, "owin")

pl.ppp<-ppp(x=x1, y=y1, window=pl.owin)

plot(pl.ppp)

pl.tes<-dirichlet(pl.ppp) # Dirichlet tessellation

a<-tile.areas(pl.tes)

a1<-a/sum(a)

ent1<-sum(-1\*a1\*log(a1))

n<-length(a)

ent.ref<-log(1/n)\*(-1)

ent.ref

ent.rel<-ent1/ent.ref # Relative H

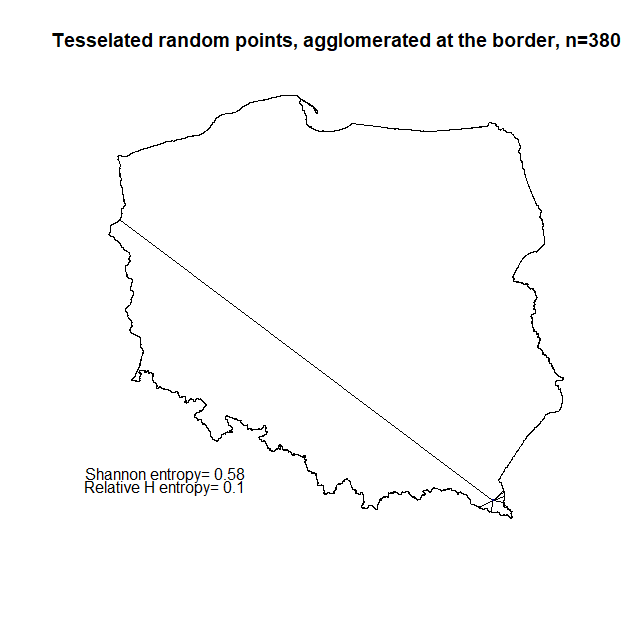
ent.rel

plot(pl.tes, main="Tesselated random points, agglomerated at the border, n=380")

plot(pl.ppp, add=TRUE, pch=".", col="darkblue", cex=2)

text(257000, 200000, paste("Shannon entropy=", round(ent1,2)))

text(257000, 180000, paste("Relative H entropy=", round(ent.rel,2)))



## # Monte Carlo simulation of relH for different locations of agglomerated points

# to prove: “*Distribution of lower bin values was conducted using Monte Carlo simulation. The whole study region was divided into 100 cells by overlaying a grid of 10 rows x 10 columns. Each cell became a limiting area for a sampling of n=380 points. Relative H varies from 0.25 to 0.47, depending on the location of agglomerated points, typically with lower values at the border and higher values inside the area.*”

bbox(pl)

# min max

#x 171677.6 861895.7

#y 133223.7 774923.7

library(raster)

r<-raster(ncols=10, nrows=10, ymn=bbox(pl)[2,1], ymx=bbox(pl)[2,2], xmn=bbox(pl)[1,1], xmx=bbox(pl)[1,2])

# numbers assigend to raster cell

r[]<-round(rep(380, times=(ncell(r)), digits=0))

plot(r, border="grey50")

# examining ID of raster cells

r[]<-1:100

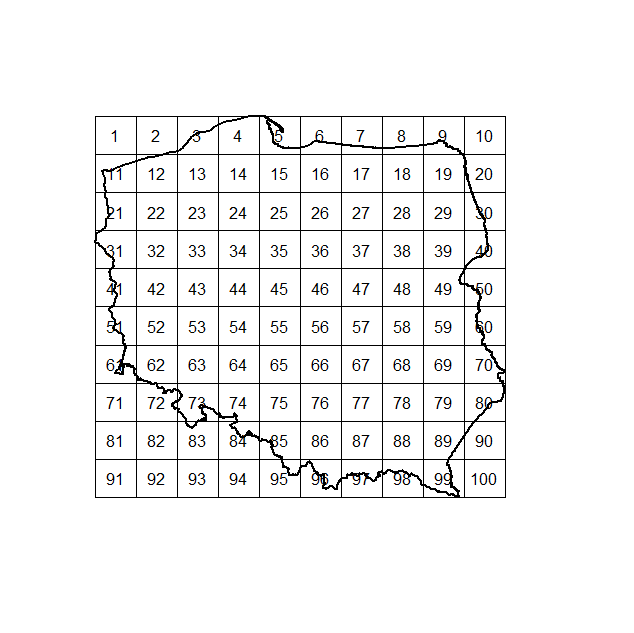
r.sp<-as(r, "SpatialPolygons")

plot(r.sp)

crds<-coordinates(r.sp)

text(crds, labels=as.numeric(1:100))

plot(pl, add=TRUE, lwd=2)



# eliminating raster cells beyond the countour

non<-c(1,2,5,4,7,8,9,10,20,71,72,81,82,83,84,90,91,92,93,94,95,100)

e<-extract(r, 1:380)

r.sp<-as(r, "SpatialPolygons")

plot(r.sp)

plot(pl, add=TRUE)

a<-crs(pl)

pl.owin<-as(pl, "owin")

r1<-r

r1[]<-rep(0, times=100)

r2<-r

r2[]<-rep(0, times=100)

all<-1:100

yes<-all[-non]

for(i in 1:length(yes)){ # loop for cells inside the countour

a1<-spsample(r.sp[yes[i]], 380, "random")

proj4string(a1)<-a

a2<-over(a1, pl)

a3<-which(a2$jpt\_nazwa\_=="POLSKA")

a4<-a1[a3,]

a4.ppp<-ppp(x=a4@coords[,1], y=a4@coords[,2], window=pl.owin)

a4.tes<-dirichlet(a4.ppp) # Dirichlet tessellation

a5<-tile.areas(a4.tes)

a6<-a5/sum(a5)

ent1<-sum(-1\*a6\*log(a6))

ent1

n<-length(a6)

ent.ref<-log(1/n)\*(-1)

ent.ref

ent.rel<-ent1/ent.ref # Relative H

ent.rel

r1[i]<-ent1 # Shannon

r2[i]<-ent.rel # Relative H

} # end of the loop

r1@data@values

r2@data@values

**# plot of Relative H on raster**

dane<-data.frame(ID=1:100)

en<-data.frame(ID=yes, RelH=r2@data@values[1:78])

dane1<-merge(dane, en, by="ID", all.x=TRUE, sort=TRUE)

dane1$RelH[is.na(dane1$RelH)==TRUE]<-0

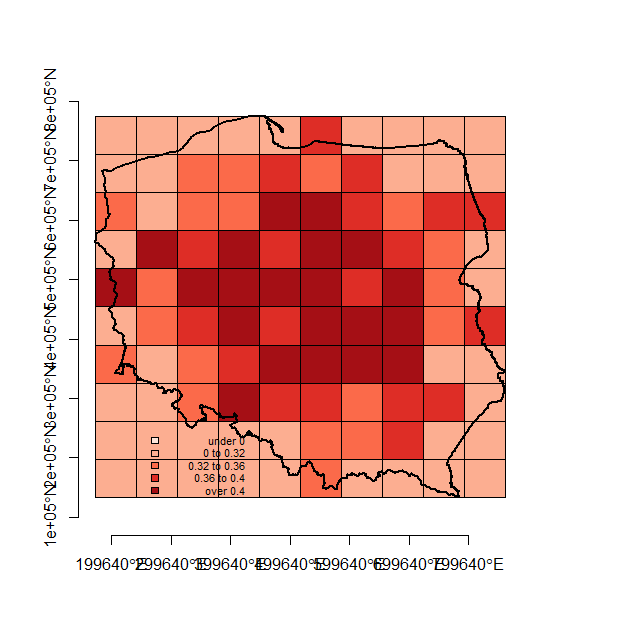
library(GISTools)

choropleth(r.sp, dane1$RelH)

plot(pl, add=TRUE, lwd=2)

odcienie<-auto.shading(dane1$RelH)

choro.legend(250000, 250000, odcienie, cex=0.65, bty="n")



# # Fig.4 Monte Carlo simulation of relH for different locations of agglomerated points

**# stratified sampling within NTS4 countours**

dane<-data.frame(ID=1:370, n=round(rnorm(370, 10, 3),0))

cords<-matrix(0, ncol=2)

a<-levels(pow$jpt\_nazwa\_)

a<-pow$jpt\_nazwa\_

for(i in 1:370){

coor<-spsample(pow[pow$jpt\_nazwa\_==a[i],], n=dane$n[i], type="random")

cords<-rbind(cords, coor@coords)}

pl.owin<-as(pl, "owin")

pl.ppp<-ppp(x=cords[,1], y=cords[,2], window=pl.owin)

pl.tes<-dirichlet(pl.ppp) # Dirichlet tessellation

a<-tile.areas(pl.tes)

a1<-a/sum(a)

ent1<-sum(-1\*a1\*log(a1))

n<-length(a)

ent.ref<-log(1/n)\*(-1)

ent.ref

ent.rel<-ent1/ent.ref # Relative H

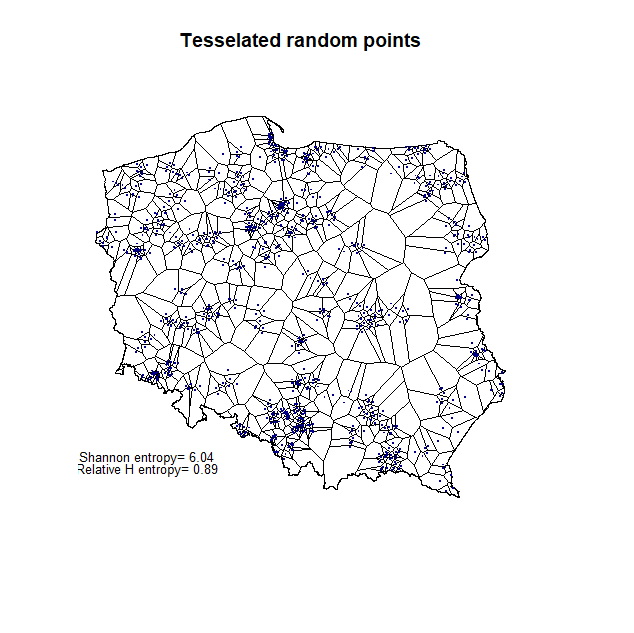
ent.rel

plot(pl.tes, main="Tesselated random points")

plot(pl.ppp, add=TRUE, pch=".", col="darkblue", cex=2)

text(260000, 200000, paste("Shannon entropy=", round(ent1,2)), cex=0.8)

text(260000, 180000, paste("Relative H entropy=", round(ent.rel,2)), cex=0.8)



**# re-shuffling locations from previous example**

wynik<-data.frame(ID=1:10, ent.ref=0, ent.rel=0)

for(i in 1:10){

coor1<-spsample(pl, n=dim(cords)[1], type="random")

pl.owin<-as(pl, "owin")

pl.ppp<-ppp(x=coor1@coords[,1], y=coor1@coords[,2], window=pl.owin)

pl.tes<-dirichlet(pl.ppp) # Dirichlet tessellation

a<-tile.areas(pl.tes)

a1<-a/sum(a)

ent1<-sum(-1\*a1\*log(a1))

n<-length(a)

ent.ref<-log(1/n)\*(-1)

ent.ref

ent.rel<-ent1/ent.ref # Relative H

ent.rel

wynik[i,2]<-ent.ref

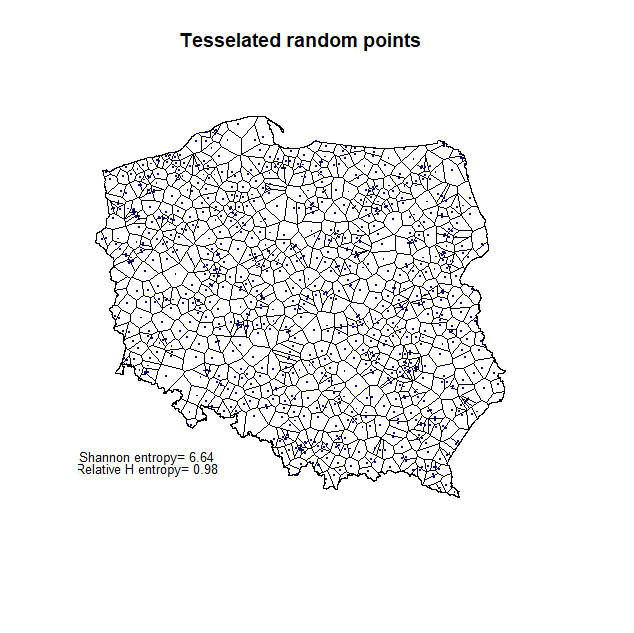
wynik[i,2]<-ent.rel}

plot(pl.tes, main="Tesselated random points")

plot(pl.ppp, add=TRUE, pch=".", col="darkblue", cex=2)

text(260000, 200000, paste("Shannon entropy=", round(ent1,2)), cex=0.8)

text(260000, 180000, paste("Relative H entropy=", round(ent.rel,2)), cex=0.8)



# # Tab.1 – repeating the excerisize by Lews (1979) - validation

theil<-matrix(0, nrow=600, ncol=4)

colnames(theil)<-c("n", "ent.empir", "ent.max", "ent.rel.H", "ent.rel.Theil")

nn<-c(15, 30, 60, 90, 180, 360) # values as in Lews (1979)

ile<-100

for(j in 1:6){

for(i in 1:ile){

b1<-spsample(pl, nn[j], "random") # CSR point-pattern

pl.owin<-as(pl, "owin")

pl.ppp<-ppp(x=b1@coords[,1], y=b1@coords[,2], window=pl.owin)

pl.tes<-dirichlet(pl.ppp) # Dirichlet tessellation

a<-tile.areas(pl.tes)

a1<-a/sum(a)

ent1<-sum(-1\*a1\*log(a1))

n<-length(a)

ent.max<-log(1/n)\*(-1)

ent.max

ent.rel<-ent1/ent.max # Relative H

ent.rel

ent.rel2<-ent.max-ent1 # Theil redundancy

ent.rel2

theil[(j-1)\*ile+i,1]<-n

theil[(j-1)\*ile+i,2]<-ent1

theil[(j-1)\*ile+i,3]<-ent.max

theil[(j-1)\*ile+i,4]<-ent.rel

theil[(j-1)\*ile+i,5]<-ent.rel2}}

theil.mean<-aggregate(theil[,4], by=list(theil[,1]), mean)

theil.sd<-aggregate(theil[,4], by=list(theil[,1]), sd)

theil.var<-aggregate(theil[,4], by=list(theil[,1]), var)

> theil.mean

Group.1 x

1 15 0.1485379

2 30 0.1557507

3 60 0.1545095

4 90 0.1474397

5 180 0.1459372

6 360 0.1451289

> theil.sd

Group.1 x

1 15 0.05664195

2 30 0.04646867

3 60 0.03460931

4 90 0.02890756

5 180 0.02146281

6 360 0.01269745

> theil.var

Group.1 x

1 15 0.0032083109

2 30 0.0021593373

3 60 0.0011978042

4 90 0.0008356467

5 180 0.0004606521

6 360 0.0001612252

## # Lews (1979) results

lews<-data.frame(n=c(15,30,60,90,180,360, 1000), theil=c(0.124, 0.129, 0.131, 0.132, 0.132, 0.132, 0.134), var=c(0.001875, 0.001225, 0.000660, 0.000412, 0.000204, 0.000090, 0.000000))

|  |  |  |
| --- | --- | --- |
| 15 | 0.124 | 0.001875 |
| 30 | 0.129 | 0.001225 |
| 60 | 0.131 | 0.000660 |
| 90 | 0.132 | 0.000412 |
| 180 | 0.132 | 0.000204 |
| 360 | 0.132 | 0.000090 |
| ∞ | 0.134 | 0.000000 |

# # Tab.2 – simulation of relH and Theil

## # CRS simulation, in PL contour

nn<-c(20, 40, 80, 160, 320, 640, 1280, 2560, 5120)

ile<-50 # number of iterations

theil<-matrix(0, nrow=12\*ile, ncol=5)

colnames(theil)<-c("n", "ent.empir", "ent.max", "ent.rel.H", "ent.rel.Theil")

for(j in 1:9){

for(i in 1:ile){

b1<-spsample(pl, nn[j], "random")

pl.owin<-as(pl, "owin")

pl.ppp<-ppp(x=b1@coords[,1], y=b1@coords[,2], window=pl.owin)

pl.tes<-dirichlet(pl.ppp) # Dirichlet tessellation

a<-tile.areas(pl.tes)

a1<-a/sum(a)

ent1<-sum(-1\*a1\*log(a1))

n<-length(a)

ent.max<-log(1/n)\*(-1)

ent.max

ent.rel<-ent1/ent.max # Relative H

ent.rel

ent.rel2<-ent.max-ent1 # Theil redundancy

ent.rel2

theil[(j-1)\*ile+i,1]<-n

theil[(j-1)\*ile+i,2]<-ent1

theil[(j-1)\*ile+i,3]<-ent.max

theil[(j-1)\*ile+i,4]<-ent.rel

theil[(j-1)\*ile+i,5]<-ent.rel2}}

theil.mean<-aggregate(theil[,3], by=list(theil[,1]), mean)

theil.sd<-aggregate(theil[,4], by=list(theil[,1]), sd)

write.table(theil, file="theil\_PL\_CSR.txt")

## # empirical data simulation – business locations in Warsaw

ROA.sp<-spTransform(ROA.sp, CRS("+proj=merc +datum=NAD83"))

waw<-spTransform(waw, CRS("+proj=merc +datum=NAD83"))

waw2<-unionSpatialPolygons(waw, IDs=rep(1, times=18)) #from maptools::

waw.owin<-as(waw2, "owin")

gg<-length(ROA.sp)

vec<-c(20, 40, 80, 160, 320, 640, 1280, 2560, 5120)

ile<-50

wynik<-matrix(0, nrow=9\*ile, ncol=5)

colnames(wynik)<-c("n", "ent.empir", "ent.max", "ent.rel.H", "ent.rel.Theil")

for(j in 1:9){

for(i in 1:50){ # zostanie do zrobienia 10 i 11

ids<-sample(1:gg, vec[j], replace=FALSE)

roa.sel<-ROA.sp[ids,]

waw.ppp<-ppp(x=roa.sel@coords[,1], y=roa.sel@coords[,2], window=waw.owin)

waw.tes<-dirichlet(waw.ppp) # Dirichlet tessellation

a<-tile.areas(waw.tes)

a1<-a/sum(a)

ent1<-sum(-1\*a1\*log(a1))

n<-length(a)

ent.max<-log(1/n)\*(-1)

ent.max

ent.rel<-ent1/ent.max # Relative H

ent.rel

ent.rel2<-ent.max-ent1 # Theil redundancy

ent.rel2

wynik[(j-1)\*ile+i,1]<-n

wynik[(j-1)\*ile+i,2]<-ent1

wynik[(j-1)\*ile+i,3]<-ent.max

wynik[(j-1)\*ile+i,4]<-ent.rel

wynik[(j-1)\*ile+i,5]<-ent.rel2}}

## # extra: theoretical simulation – random point within box

**# box**

x1<-c(0,2,2,0)

y1<-c(0, 0,2,2)

xy1<-SpatialPointsDataFrame(matrix(c(x1,y1), ncol=2), data.frame(ID=seq(1:length(x1))),

proj4string=CRS("+proj=longlat +ellps=WGS84 +datum=WGS84"))

xy1.m<-matrix(c(x1,y1), ncol=2)

xy1.m<-rbind(xy1.m[dim(xy1.m)[1],],xy1.m) # one extra row

xy1.poly<-spPolygons(xy1.m)

plot(xy1.poly)

axis(1)

axis(2)

proj4string(xy1.poly)<-"+proj=longlat +datum=WGS84 +ellps=WGS84"

xy.poly.merc<- spTransform(xy1.poly, CRS("+proj=merc +datum=NAD83"))

plot(xy.poly.merc)

axis(1)

axis(2)

theil<-matrix(0, nrow=65, ncol=4)

colnames(theil)<-c("n", "ent.empir", "ent.max", "ent.rel.H", "ent.rel.Theil")

nn<-c(15, 30, 60, 120, 240, 480, 960, 1920, 3840, 7680, 15360, 30720)

for(j in 1:12){

for(i in 1:5){

**# random pattern**

b<-spsample(xy.poly.merc, n=nn[j], "**random**")

box.owin<-as(xy.poly.merc, "owin")

box.ppp<-ppp(x=b@coords[,1], y=b@coords[,2], window=box.owin)

box.tes<-dirichlet(box.ppp) # Dirichlet tessellation

a<-tile.areas(box.tes)

a1<-a/sum(a)

ent1<-sum(-1\*a1\*log(a1))

n<-length(a)

ent.max<-log(1/n)\*(-1)

ent.max

ent.rel<-ent1/ent.max # Relative H

ent.rel

ent.rel2<-ent.max-ent1 # Theil redundancy

ent.rel2

theil[(j-1)\*5+i,1]<-n

theil[(j-1)\*5+i,2]<-ent1

theil[(j-1)\*5+i,3]<-ent.max

theil[(j-1)\*5+i,4]<-ent.rel

theil[(j-1)\*5+i,5]<-ent.rel2}}

theil.mean<-aggregate(theil[,3], by=list(theil[,1]), mean)

theil.sd<-aggregate(theil[,4], by=list(theil[,1]), sd)

theil

n ent.max ent.rel.H ent.rel.Theil

[1,] 15 2.708050 0.9532534 0.12659206

[2,] 15 2.708050 0.9541282 0.12422322

[3,] 15 2.708050 0.8967229 0.27967962

[4,] 15 2.708050 0.9607522 0.10628499

[5,] 15 2.708050 0.9001109 0.27050479

[6,] 30 3.401197 0.9647489 0.11989589

[7,] 30 3.401197 0.9448609 0.18753894

[8,] 30 3.401197 0.9378078 0.21152803

[9,] 30 3.401197 0.9656665 0.11677516

[10,] 30 3.401197 0.9581399 0.14237452

[11,] 60 4.094345 0.9608315 0.16036954

[12,] 60 4.094345 0.9637623 0.14836977

[13,] 60 4.094345 0.9691146 0.12645549

[14,] 60 4.094345 0.9702459 0.12182358

[15,] 60 4.094345 0.9765692 0.09593388

[16,] 120 4.787492 0.9670234 0.15787505

[17,] 120 4.787492 0.9704437 0.14150035

[18,] 120 4.787492 0.9674834 0.15567303

[19,] 120 4.787492 0.9741143 0.12392774

[20,] 120 4.787492 0.9634157 0.17514713

[21,] 240 5.480639 0.9707126 0.16051368

[22,] 240 5.480639 0.9680227 0.17525620

[23,] 240 5.480639 0.9777289 0.12205977

[24,] 240 5.480639 0.9712972 0.15730967

[25,] 240 5.480639 0.9719214 0.15388878

[26,] 480 6.173786 0.9763923 0.14574874

[27,] 480 6.173786 0.9777608 0.13730018

[28,] 480 6.173786 0.9748491 0.15527599

[29,] 480 6.173786 0.9748464 0.15529295

[30,] 480 6.173786 0.9777236 0.13752964

[31,] 960 6.866933 0.9820664 0.12314892

[32,] 960 6.866933 0.9797117 0.13931837

[33,] 960 6.866933 0.9810915 0.12984353

[34,] 960 6.866933 0.9814719 0.12723156

[35,] 960 6.866933 0.9787493 0.14592740

[36,] 1920 7.560080 0.9815528 0.13946262

[37,] 1920 7.560080 0.9819996 0.13608418

[38,] 1920 7.560080 0.9818183 0.13745501

[39,] 1920 7.560080 0.9814745 0.14005429

[40,] 1920 7.560080 0.9824667 0.13255334

[41,] 3840 8.253228 0.9832161 0.13852117

[42,] 3840 8.253228 0.9834705 0.13642140

[43,] 3840 8.253228 0.9838513 0.13327875

[44,] 3840 8.253228 0.9829650 0.14059371

[45,] 3840 8.253228 0.9831082 0.13941201

[46,] 7680 8.946375 0.9848240 0.13577059

[47,] 7680 8.946375 0.9847297 0.13661390

[48,] 7680 8.946375 0.9844350 0.13925037

[49,] 7680 8.946375 0.9850711 0.13355972

[50,] 7680 8.946375 0.9848598 0.13545032

[51,] 15360 9.639522 0.9856967 0.13787667

[52,] 15360 9.639522 0.9859646 0.13529486

[53,] 15360 9.639522 0.9859498 0.13543703

[54,] 15360 9.639522 0.9856279 0.13854065

[55,] 15360 9.639522 0.9862511 0.13253293

[56,] 30720 10.332669 0.9868882 0.13548001

[57,] 30720 10.332669 0.9871582 0.13269027

[58,] 30720 10.332669 0.9870976 0.13331669

[59,] 30720 10.332669 0.9867699 0.13670268

[60,] 30720 10.332669 0.9867001 0.13742365

# # Business locations in Warsaw city

## # Fig.6a - Entropy for tesselation of empirical data

ROA.sp<-spTransform(ROA.sp, CRS("+proj=merc +datum=NAD83"))

waw<-spTransform(waw, CRS("+proj=merc +datum=NAD83"))

waw2<-unionSpatialPolygons(waw, IDs=rep(1, times=18)) #with maptools::

plot(waw2, lwd=2)

plot(waw, add=TRUE, border="red")

waw.owin<-as(waw2, "owin")

waw.ppp<-ppp(x=ROA.sp@coords[,1], y=ROA.sp@coords[,2], window=waw.owin)

waw.tes<-dirichlet(waw.ppp) # Dirichlet tessellation

a<-tile.areas(waw.tes)

a1<-a/sum(a)

ent1<-sum(-1\*a1\*log(a1))

n<-length(a)

ent.ref<-log(1/n)\*(-1)

ent.ref

ent.rel<-ent1/ent.ref # Relative H

ent.rel

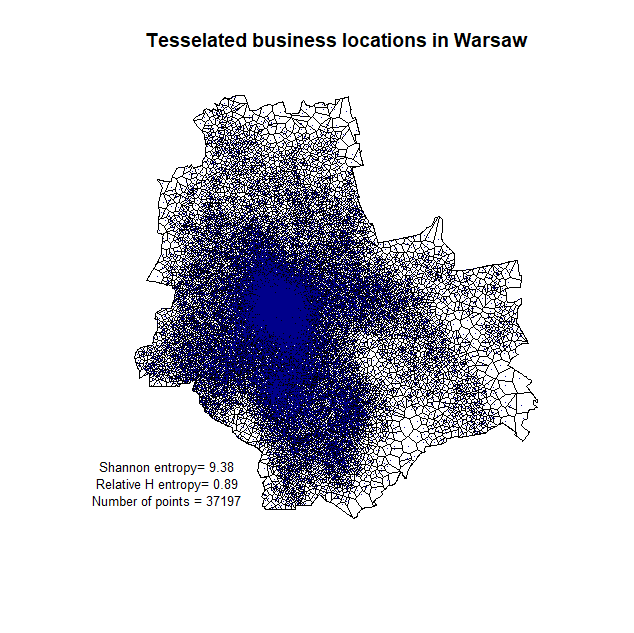
plot(waw.tes, main="Tesselated business locations in Warsaw")

plot(waw.ppp, add=TRUE, pch=".", col="darkblue", cex=0.8)

text(2325000, 6790000, paste("Shannon entropy=", round(ent1,2)), cex=0.8)

text(2325000, 6788000, paste("Relative H entropy=", round(ent.rel,2)), cex=0.8)

text(2325000, 6786000, paste("Number of points =", round(n,2)), cex=0.8)



## # Fig.3 – Normality of Shannon’s H distribution

ROA.sp<-spTransform(ROA.sp, CRS("+proj=merc +datum=NAD83"))

waw<-spTransform(waw, CRS("+proj=merc +datum=NAD83"))

waw2<-unionSpatialPolygons(waw, IDs=rep(1, times=18)) # maptools::

plot(waw2, lwd=2)

plot(waw, add=TRUE, border="red")

waw.owin<-as(waw2, "owin")

gg<-length(ROA.sp)

vec<-c(15,30,60,90,180,360,720,1440,2880,5760)

wynik<-matrix(0, nrow=100, ncol=10)

colnames(wynik)<-vec

for(i in 1:10){

for(j in 1:100){

ids<-sample(1:gg, vec[i], replace=FALSE)

roa.sel<-ROA.sp[ids,]

waw.ppp<-ppp(x=roa.sel@coords[,1], y=roa.sel@coords[,2], window=waw.owin)

waw.tes<-dirichlet(waw.ppp) # Dirichlet tessellation

a<-tile.areas(waw.tes)

a1<-a/sum(a)

ent1<-sum(-1\*a1\*log(a1))

wynik[j,i]<-round(ent1, 2)}}

plot(density(scale(wynik[,1])), lty=1, lwd=1, xlim=c(-5, 5), ylim=c(0,0.5), main="Normalised distributions of H(empir) for k=100 iterations

n=15, 30, 60, 90, 180, 360, 720, 1440, 2880, 5760", sub="Normal distribution N(0,1) in red")

lines(density(scale(wynik[,2])), lty=1, lwd=2)

lines(density(scale(wynik[,3])), lty=1, lwd=3)

lines(density(scale(wynik[,4])), lty=2, lwd=1)

lines(density(scale(wynik[,5])), lty=2, lwd=2)

lines(density(scale(wynik[,6])), lty=2, lwd=3)

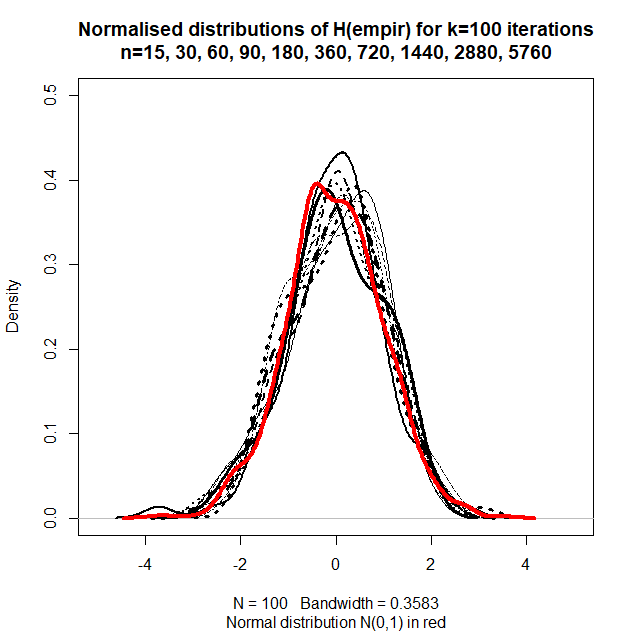
lines(density(scale(wynik[,7])), lty=3, lwd=1)

lines(density(scale(wynik[,8])), lty=3, lwd=2)

lines(density(scale(wynik[,9])), lty=3, lwd=3)

lines(density(scale(wynik[,10])), lty=4, lwd=1)

lines(density(rnorm(1000, 0,1)), lty=1, lwd=4, col="red")



apply(wynik, 2, mean)

apply(wynik, 2, sd)

shapiro.test(wynik[,1])

testy<-matrix(0,ncol=2, nrow=10)

for(i in 1:10){

a<-shapiro.test(wynik[,i])

testy[i,1]<-a$statistic

testy[i,2]<-a$p.value

}

## # extra: simulation – relative entropy for various sample size n

ROA.sp<-spTransform(ROA.sp, CRS("+proj=merc +datum=NAD83"))

waw<-spTransform(waw, CRS("+proj=merc +datum=NAD83"))

waw2<-unionSpatialPolygons(waw, IDs=rep(1, times=18)) # maptools

plot(waw2, lwd=2)

plot(waw, add=TRUE, border="red")

waw.owin<-as(waw2, "owin")

gg<-length(ROA.sp)

vec<-c(15,30,60,90,180,360,720,1440,2880,5760, 11520, 23040)

wynik<-matrix(0, nrow=12, ncol=5)

colnames(wynik)<-c("Shannon entropy", "Max entropy", "Relative H entropy", "Theil’s redundancy", "n")

for(i in 10:12){

ids<-sample(1:gg, vec[i], replace=FALSE)

roa.sel<-ROA.sp[ids,]

waw.ppp<-ppp(x=roa.sel@coords[,1], y=roa.sel@coords[,2], window=waw.owin)

waw.tes<-dirichlet(waw.ppp) # Dirichlet tessellation

a<-tile.areas(waw.tes)

a1<-a/sum(a)

ent1<-sum(-1\*a1\*log(a1))

n<-length(a)

ent.max<-log(1/n)\*(-1)

ent.max

ent.rel<-ent1/ent.max2 # Relative H

ent.rel

ent.rel2<-ent.max2-ent1 # Theil’s redundancy

ent.rel2

plot(waw.tes, main="Tesselated business locations in Warsaw")

plot(waw.ppp, add=TRUE, pch=".", col="darkblue", cex=0.8)

text(2325000, 6790000, paste("Shannon entropy=", round(ent1,2)), cex=0.8)

text(2325000, 6788000, paste("Relative H entropy=", round(ent.rel,2)), cex=0.8)

text(2325000, 6786000, paste("Theil’s redundancy=", round(ent.rel2,2)), cex=0.8)

text(2325000, 6784000, paste("Number of points=", round(n,2)), cex=0.8)

wynik[i,1]<-round(ent1, 5)

wynik[i,2]<-round(ent.max2, 5)

wynik[i,3]<-round(ent.rel, 5)

wynik[i,4]<-round(ent.rel2, 5)

wynik[i,5]<-round(n, 5)}

wynik

Shannon entropy Max entropy Relative H entropy Theil’s redundancy n

[1,] 2.28985 2.70805 0.84557 0.41820 15

[2,] 2.52729 3.40120 0.74306 0.87391 30

[3,] 3.23825 4.09434 0.79091 0.85609 60

[4,] 3.70973 4.49981 0.82442 0.79008 90

[5,] 4.30523 5.19296 0.82905 0.88772 180

[6,] 5.06949 5.88610 0.86126 0.81661 360

[7,] 5.58541 6.57925 0.84894 0.99385 720

[8,] 6.26449 7.27240 0.86141 1.00791 1440

[9,] 6.94354 7.96555 0.87170 1.02201 2880

[10,] 7.58001 8.65869 0.87542 1.07868 5760

[11,] 8.26180 9.35184 0.88344 1.09004 11520

[12,] 8.92612 10.04499 0.88861 1.11887 23040

# # Fig.6b - Analysis of gridded population in Warsaw city

**# operations on grid for population**

# limiting grid with contour of Warsaw city

lim<-over(pop.grid, waw.union)

pop.grid.waw<-pop.grid[lim==1, ]

a<-which(lim==1)

pop.df.waw<-pop.df[a,]

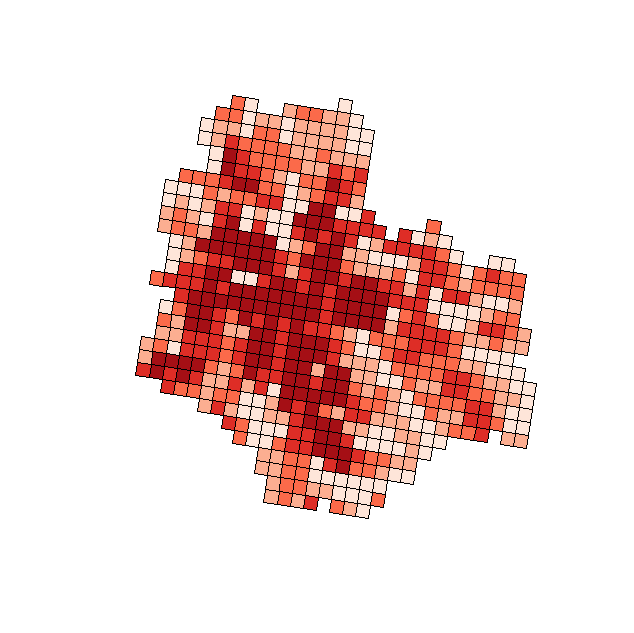
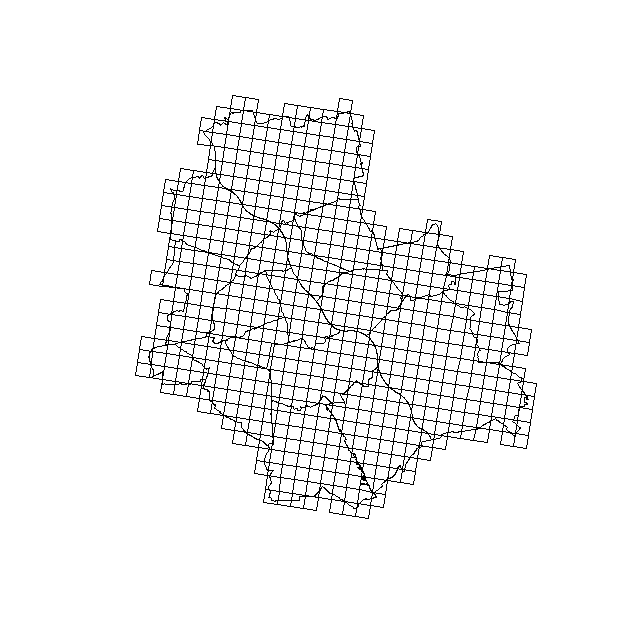
plot(pop.grid.waw) # plotting limited grid

plot(waw, add=TRUE)

library(GISTools) # plotting limited grid and population

choropleth(pop.grid.waw, pop.df.waw$TOT)

pop.merc<-spTransform(pop, CRS("+proj=merc +datum=NAD83"))



**# generating points within grid cells which represent number of people**

a<-which(pop.df.waw$TOT<100)

pop.df.waw$TOT[a]<-100

pts<-data.frame(x=0, y=0)

for(i in 1:601){

probka<-spsample(pop.grid.waw[i,], pop.df.waw$TOT[i]/100, type="random")

probka.df<-as.data.frame(probka@coords)

colnames(probka.df)<-c("x", "y")

pts<-rbind(pts, probka.df)}

pts<-pts[-1,]

plot(waw)

plot(pop.grid.waw, add=TRUE, border="grey80")

points(pts, pch=".")

pts1<-pts

pts.sp<-pts

coordinates(pts.sp)<-c("x", "y")

proj4string(pts.sp)<-CRS("+proj=longlat +datum=NAD83")

pts.lim<-over(pts.sp, waw.union)

a2<-which(is.na(pts.lim)==TRUE)

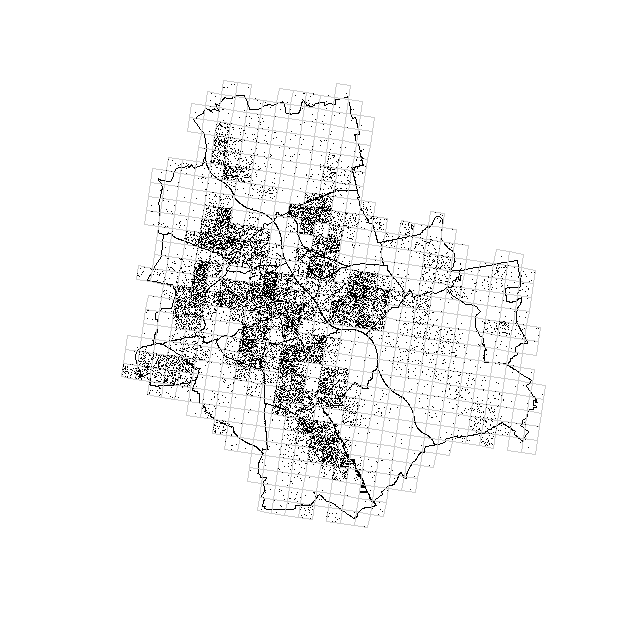
pts1<-pts1[-a2,]

plot(waw)

plot(pop.grid.waw, add=TRUE, border="grey80")

points(pts1, pch=".")

pts1.sp<-pts1



# changing the projection from spherical to planar

coordinates(pts1.sp)<-c("x", "y")

proj4string(pts1.sp)<-CRS("+proj=longlat +datum=NAD83")

pts1.sp<-spTransform(pts1.sp, CRS("+proj=merc +datum=NAD83"))

**# Shannon H for city on pointed population**

waw.merc<-spTransform(waw, CRS("+proj=merc +datum=NAD83"))

waw2<-unionSpatialPolygons(waw.merc, IDs=rep(1, times=18)) #maptools

plot(waw2, lwd=2)

plot(waw, add=TRUE, border="red")

waw.owin<-as(waw2, "owin")

waw.ppp<-ppp(x=pts1.sp@coords[,1], y=pts1.sp@coords[,2], window=waw.owin)

waw.tes<-dirichlet(waw.ppp) # Dirichlet tessellation

a<-tile.areas(waw.tes)

a1<-a/sum(a)

ent1<-sum(-1\*a1\*log(a1))

n<-length(a)

ent.ref<-log(1/n)\*(-1)

ent.ref

ent.rel<-ent1/ent.ref # Relative H

ent.rel

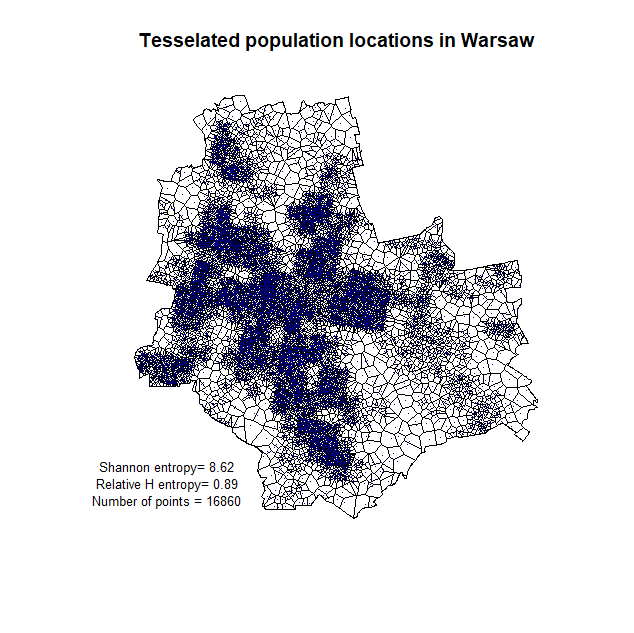
plot(waw.tes, main="Tesselated population locations in Warsaw")

plot(waw.ppp, add=TRUE, pch=".", col="darkblue", cex=0.8)

text(2325000, 6790000, paste("Shannon entropy=", round(ent1,2)), cex=0.8)

text(2325000, 6788000, paste("Relative H entropy=", round(ent.rel,2)), cex=0.8)

text(2325000, 6786000, paste("Number of points =", round(n,2)), cex=0.8)



# # Fig.5b – Business location in Lubelskie region

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

firmy.sp<-firmy

coordinates(firmy.sp)<-c("coords.x1", "coords.x2")

proj4string(firmy.sp)<-CRS("+proj=longlat +datum=NAD83")

firmy.sp<-spTransform(firmy.sp, CRS("+proj=merc +datum=NAD83"))

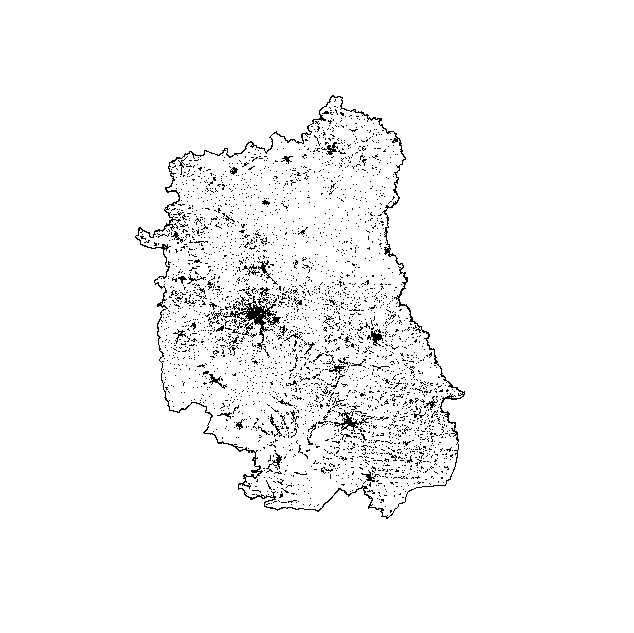
woj<-readOGR(".", "wojewodztwa") # 16 jedn.

woj<- spTransform(woj, CRS("+proj=merc +datum=NAD83"))

lub<-woj[woj@data$jpt\_nazwa\_=="lubelskie",]

plot(woj[woj@data$jpt\_nazwa\_=="lubelskie",])

points(firmy.sp, pch=".")



lub.owin<-as(lub, "owin")

lub.ppp<-ppp(x=firmy.sp@coords[,1], y=firmy.sp@coords[,2], window=lub.owin)

lub.tes<-dirichlet(lub.ppp) # Dirichlet tessellation

a<-tile.areas(lub.tes)

a1<-a/sum(a)

ent1<-sum(-1\*a1\*log(a1))

n<-length(a)

ent.ref<-log(1/n)\*(-1)

ent.ref

ent.rel<-ent1/ent.ref # Relative H

ent.rel

plot(lub.tes, main="Tesselated business locations in lubelskie region

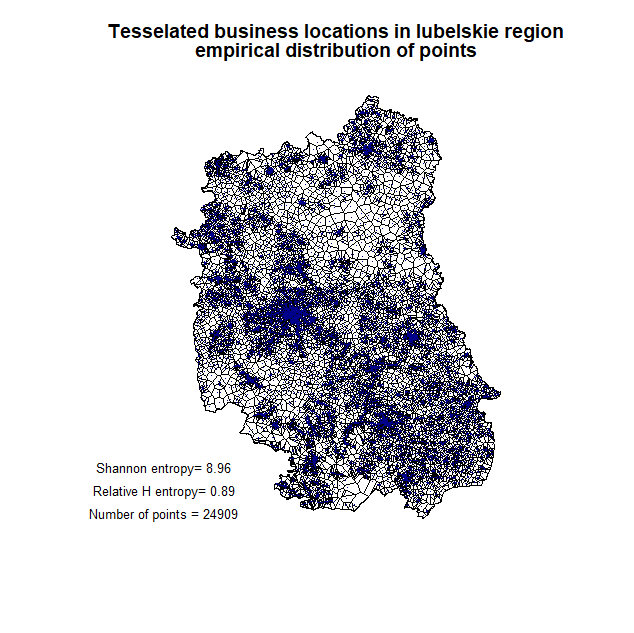
empirical distribution of points")

plot(lub.ppp, add=TRUE, pch=".", col="darkblue", cex=0.8)

text(2400000, 6500000, paste("Shannon entropy=", round(ent1,2)), cex=0.8)

text(2400000, 6480000, paste("Relative H entropy=", round(ent.rel,2)), cex=0.8)

text(2400000, 6460000, paste("Number of points =", round(n,2)), cex=0.8)



# # Fig.5a –Business locations in Silesian region

silesia<-read.csv("dane\_slaskie\_FULL.csv", header=TRUE, dec=".", sep=",")

library(doBy)

silesia$los<-runif(dim(silesia)[1], 0,1)

silesia<-orderBy(~los, silesia)

sil<-silesia[1:25000,]

sil.sp<-sil

coordinates(sil.sp)<-c("coords.x1", "coords.x2")

proj4string(sil.sp)<-CRS("+proj=longlat +datum=NAD83")

sil.sp<-spTransform(sil.sp, CRS("+proj=merc +datum=NAD83"))

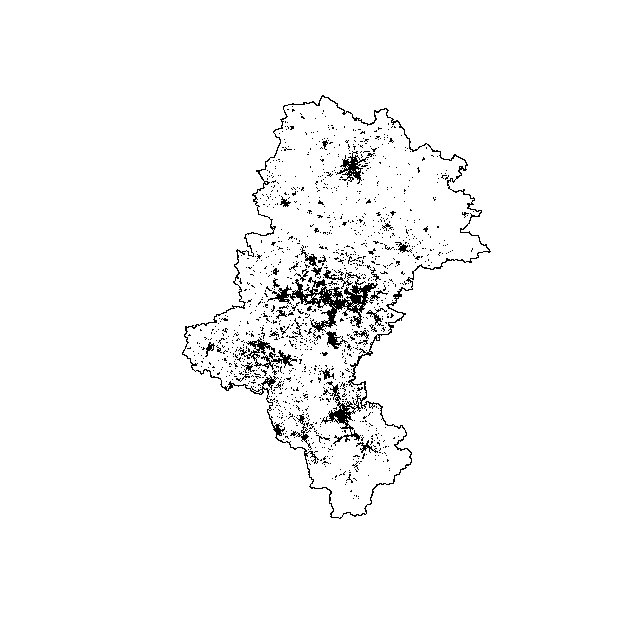
woj<-readOGR(".", "wojewodztwa") # 16 jedn.

woj<- spTransform(woj, CRS("+proj=merc +datum=NAD83"))

sla<-woj[woj@data$jpt\_nazwa\_=="śląskie",]

plot(sla)

points(sil.sp, pch=".")



sla.owin<-as(sla, "owin")

sla.ppp<-ppp(x=sil.sp@coords[,1], y=sil.sp@coords[,2], window=sla.owin)

sla.tes<-dirichlet(sla.ppp) # Dirichlet tessellation

a<-tile.areas(sla.tes)

a1<-a/sum(a)

ent1<-sum(-1\*a1\*log(a1))

n<-length(a)

ent.ref<-log(1/n)\*(-1)

ent.ref

ent.rel<-ent1/ent.ref # Relative H

ent.rel

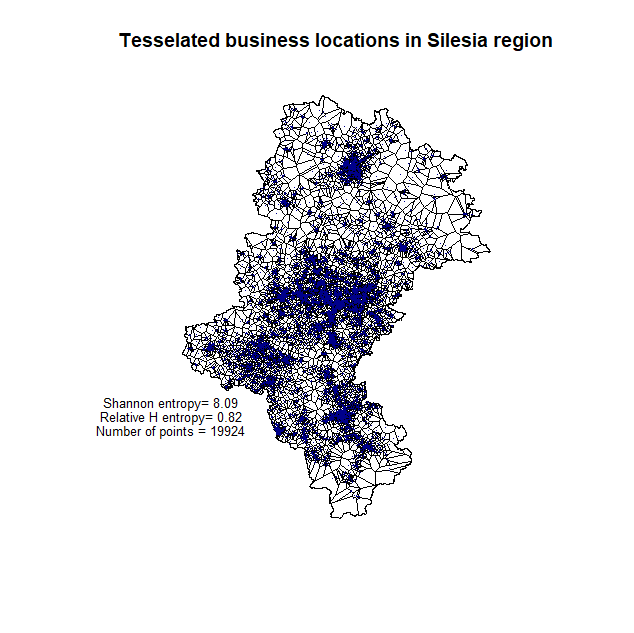
plot(sla.tes, main="Tesselated business locations in Silesia region")

plot(sla.ppp, add=TRUE, pch=".", col="darkblue", cex=0.8)

text(2000000, 6390000, paste("Shannon entropy=", round(ent1,2)), cex=0.8)

text(2000000, 6380000, paste("Relative H entropy=", round(ent.rel,2)), cex=0.8)

text(2000000, 6370000, paste("Number of points =", round(n,2)), cex=0.8)



# # Fig.6c – Business location in Silesian conurbation (city)

silesia<-read.csv("dane\_slaskie\_FULL.csv", header=TRUE, dec=".", sep=",")

**# selecting NTS4 to create conurbation – mentropolitan area**

#levels(silesia$powiatowe)

# [1] "powiat będziński" "powiat bielski"

# [3] "powiat Bielsko-Biała" "powiat bieruńsko-lędziński"

# [5] "powiat Bytom" "powiat Chorzów"

# [7] "powiat cieszyński" "powiat Częstochowa"

# [9] "powiat częstochowski" "powiat Dąbrowa Górnicza"

#[11] "powiat Gliwice" "powiat gliwicki"

#[13] "powiat Jastrzębie-Zdrój" "powiat Jaworzno"

#[15] "powiat Katowice" "powiat kłobucki"

#[17] "powiat lubliniecki" "powiat mikołowski"

#[19] "powiat Mysłowice" "powiat myszkowski"

#[21] "powiat Piekary Śląskie" "powiat pszczyński"

#[23] "powiat raciborski" "powiat Ruda Śląska"

#[25] "powiat rybnicki" "powiat Rybnik"

#[27] "powiat Siemianowice Śląskie" "powiat Sosnowiec"

#[29] "powiat Świętochłowice" "powiat tarnogórski"

#[31] "powiat Tychy" "powiat wodzisławski"

#[33] "powiat Zabrze" "powiat zawierciański"

#[35] "powiat Żory" "powiat żywiecki"

dane.konur<-silesia[silesia$powiatowe=="powiat Gliwice" | silesia$powiatowe=="powiat Zabrze" | silesia$powiatowe=="powiat Katowice" | silesia$powiatowe=="powiat Bytom" | silesia$powiatowe=="powiat Świętochłowice" | silesia$powiatowe=="powiat Siemianowice Śląskie" | silesia$powiatowe=="powiat Sosnowiec" | silesia$powiatowe=="powiat Dąbrowa Górnicza" | silesia$powiatowe=="powiat Jaworzno" | silesia$powiatowe=="powiat Mysłowice" | silesia$powiatowe=="powiat będziński" | silesia$powiatowe=="powiat Tychy" | silesia$powiatowe=="powiat Ruda Śląska" | silesia$powiatowe=="powiat Piekary Śląskie" | silesia$powiatowe=="powiat Chorzów" | silesia$powiatowe=="powiat mikołowski" | silesia$powiatowe=="powiat tarnogórski", ]

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

pow<-readOGR(".", "powiaty") # 16 jedn.

gm<-readOGR(".", "gminy2487\_2005") # 16 jedn.

pow<-spTransform(pow, CRS("+proj=longlat +datum=NAD83"))

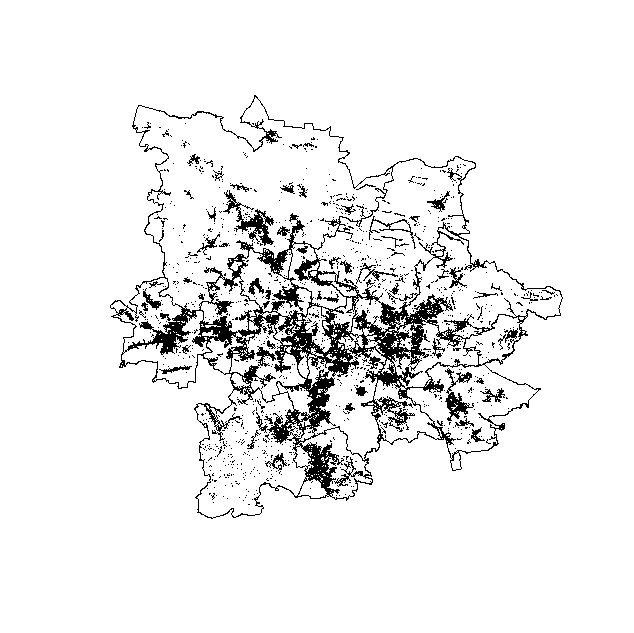
gm<-spTransform(gm, CRS("+proj=longlat +datum=NAD83"))

#levels(pow@data@jpt\_nazwa\_)

pow.konur<-pow[pow@data$jpt\_nazwa\_=="powiat Gliwice" | pow@data$jpt\_nazwa\_=="powiat Zabrze" | pow@data$jpt\_nazwa\_=="powiat Katowice" | pow@data$jpt\_nazwa\_=="powiat Bytom" | pow@data$jpt\_nazwa\_=="powiat Świętochłowice" | pow@data$jpt\_nazwa\_=="powiat Siemianowice Śląskie" | pow@data$jpt\_nazwa\_=="powiat Sosnowiec" | pow@data$jpt\_nazwa\_=="powiat Dąbrowa Górnicza" | pow@data$jpt\_nazwa\_=="powiat Jaworzno" | pow@data$jpt\_nazwa\_=="powiat Mysłowice" | pow@data$jpt\_nazwa\_=="powiat będziński" | pow@data$jpt\_nazwa\_=="powiat Tychy" | pow@data$jpt\_nazwa\_=="powiat Ruda Śląska" | pow@data$jpt\_nazwa\_=="powiat Piekary Śląskie" | pow@data$jpt\_nazwa\_=="powiat Chorzów" | pow@data$jpt\_nazwa\_=="powiat mikołowski" | pow@data$jpt\_nazwa\_=="powiat tarnogórski", ]

plot(pow.konur)

points(dane.konur[,17:18], pch=".")



**# selecting NTS5 units to create metropolitan area**

g.konur<-over(gm, pow.konur)

a<-which(g.konur$iip\_przest=="PL.PZGIK.200")

gm.konur<-gm[a,]

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

gm.konur.df<-gm.df[a,]

crds<-coordinates(gm.konur)

plot(gm.konur)

text(crds, labels=gm.konur.df$NAZWA.1, cex=0.5)

gm.konur<-gm[gm.df$NAZWA.1=="BĘDZIN" | gm.df$NAZWA.1=="BYTOM" | gm.df$NAZWA.1=="CHORZÓW" | gm.df$NAZWA.1=="CZELADŹ" | gm.df$NAZWA.1=="DĄBROWA GÓRNICZA" | gm.df$NAZWA.1=="GLIWICE" | gm.df$NAZWA.1=="JAWORZNO" | gm.df$NAZWA.1=="KATOWICE" | gm.df$NAZWA.1=="MIKOŁÓW" | gm.df$NAZWA.1=="PIEKARY ŚLĄSKIE" | gm.df$NAZWA.1=="RUDA ŚLĄSKA" | gm.df$NAZWA.1=="SIEMIANOWICE ŚLĄSKIE" | gm.df$NAZWA.1=="SOSNOWIEC" | gm.df$NAZWA.1=="TARNOWSKIE GÓRY" | gm.df$NAZWA.1=="TYCHY" | gm.df$NAZWA.1=="ZABRZE" | gm.df$NAZWA.1=="ŚWIĘTOCHŁOWICE" | gm.df$NAZWA.1=="MYSŁOWICE" | gm.df$NAZWA.1=="ŚWIERKLANIEC", ]

konur.union<-unionSpatialPolygons(gm.konur, IDs=rep(1, times=19)) #maptools

konur.union<-spTransform(konur.union, CRS("+proj=merc +datum=NAD83"))

dane.konur.sp<-dane.konur

a<-which(is.na(dane.konur.sp$coords.x1)==TRUE)

dane.konur.sp<-dane.konur.sp[-a,]

dane.konur.sp$coords.x1<-dane.konur.sp$coords.x1+rnorm(dim(dane.konur.sp)[1], mean=0, sd=0.01)

dane.konur.sp$coords.x2<-dane.konur.sp$coords.x2+rnorm(dim(dane.konur.sp)[1], mean=0, sd=0.01)

coordinates(dane.konur.sp)<-c("coords.x1", "coords.x2")

proj4string(dane.konur.sp)<-CRS("+proj=longlat +datum=NAD83")

dane.konur.sp<-spTransform(dane.konur.sp, CRS("+proj=merc +datum=NAD83"))

dane.lim<-over(dane.konur.sp, konur.union)

a<-which(is.na(dane.lim)==TRUE)

dane.konur.sp<-dane.konur.sp[-a,]

plot(konur.union)

points(dane.konur.sp, pch=".")

library(doBy)

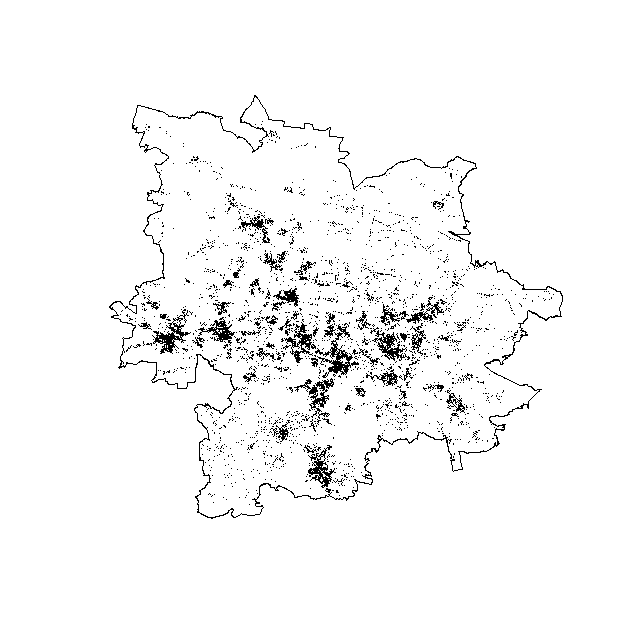
dane.konur.sp$los<-runif(dim(dane.konur.sp)[1], 0,1)

dane.konur.sp<-orderBy(~los, dane.konur.sp)

sub.konur.sp<-dane.konur.sp[1:25000,]

plot(konur.union)

points(sub.konur.sp, pch=".")



**# calculating Entropy for business location**

konur.owin<-as(konur.union, "owin")

konur.ppp<-ppp(x=sub.konur.sp@coords[,1], y=sub.konur.sp@coords[,2], window=konur.owin)

konur.tes<-dirichlet(konur.ppp) # Dirichlet tessellation

a<-tile.areas(konur.tes)

a1<-a/sum(a)

ent1<-sum(-1\*a1\*log(a1))

n<-length(a)

ent.ref<-log(1/n)\*(-1)

ent.ref

ent.rel<-ent1/ent.ref # Relative H

ent.rel

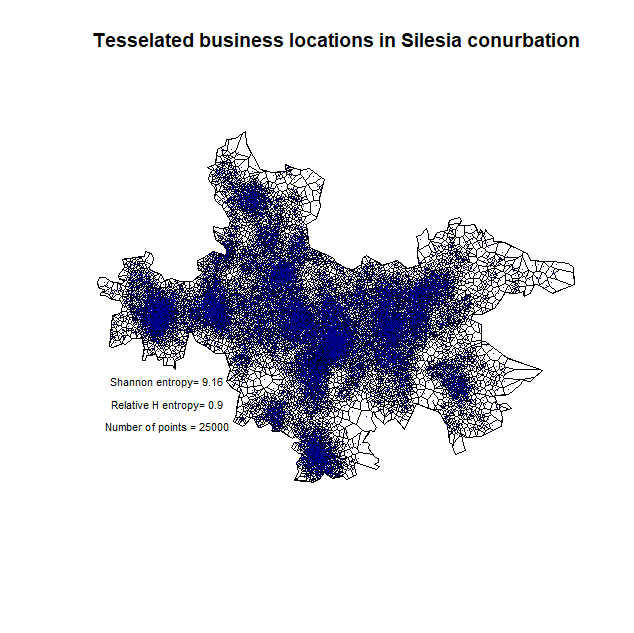
plot(konur.tes, main="Tesselated business locations in Silesia conurbation", xlim=c(2040000, 2160000))

plot(konur.ppp, add=TRUE, pch=".", col="darkblue", cex=0.8)

text(2080000, 6450000, paste("Shannon entropy=", round(ent1,2)), cex=0.7)

text(2080000, 6445000, paste("Relative H entropy=", round(ent.rel,2)), cex=0.7)

text(2080000, 6440000, paste("Number of points =", round(n,2)), cex=0.7)



# # Fig.6d - Analysis of gridded population in Silesian metropolitan area

**# initial operations on grid**

lim<-over(pop.grid, konur.union)

pop.grid.sil<-pop.grid[lim==1, ]

a<-which(lim==1)

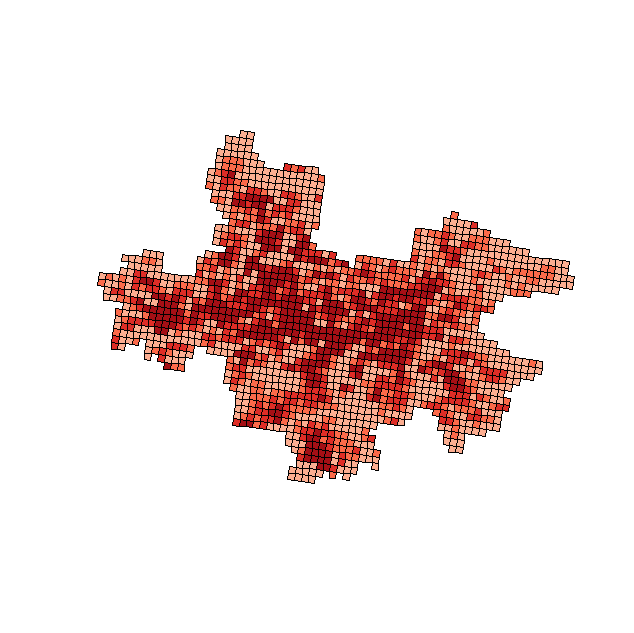
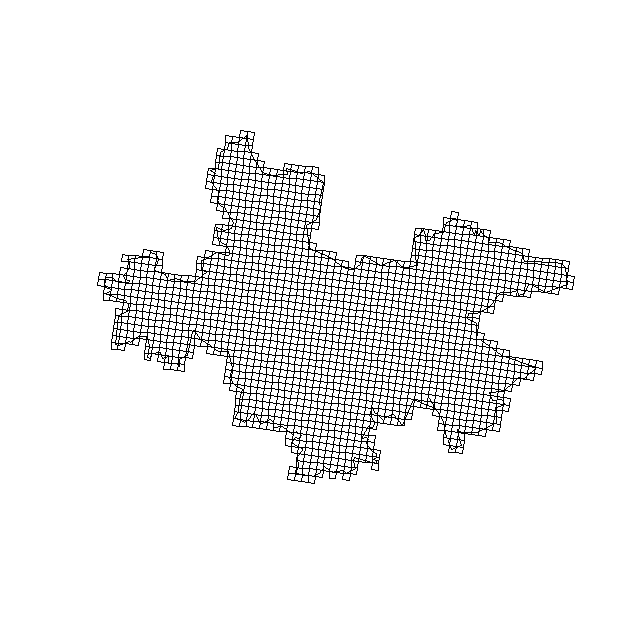
pop.df.sil<-pop.df[a,]

plot(pop.grid.sil)

plot(konur.union, add=TRUE)

library(GISTools)

choropleth(pop.grid.sil, pop.df.sil$TOT)



**# sampling points within grid cells**

a<-which(pop.df.sil$TOT<100)

pop.df.sil$TOT[a]<-100

pts<-data.frame(x=0, y=0)

for(i in 1:1671){

probka<-spsample(pop.grid.sil[i,], pop.df.sil$TOT[i]/100, type="random", iter=10)

probka.df<-as.data.frame(probka@coords)

colnames(probka.df)<-c("x", "y")

pts<-rbind(pts, probka.df)}

pts<-pts[-1,]

plot(konur.union)

plot(pop.grid.sil, add=TRUE, border="grey80")

points(pts, pch=".")

pts1<-pts

pts.sp<-pts

coordinates(pts.sp)<-c("x", "y")

proj4string(pts.sp)<-CRS("+proj=merc +datum=NAD83")

pts.lim<-over(pts.sp, konur.union)

a2<-which(is.na(pts.lim)==TRUE)

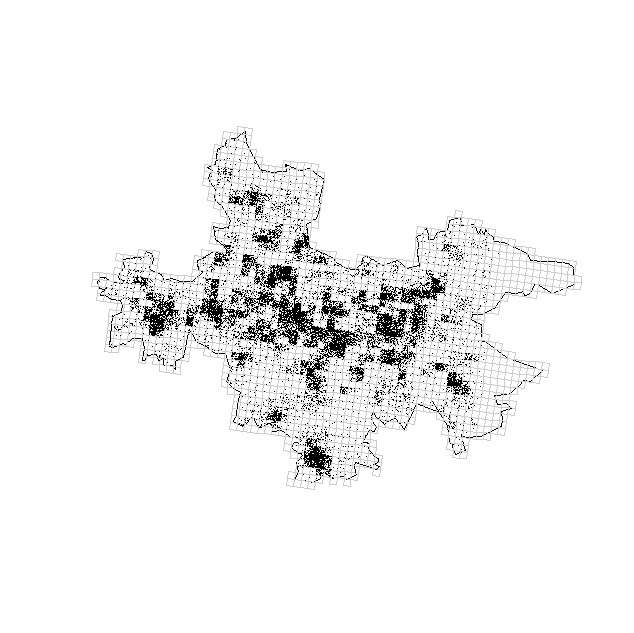
pts1<-pts1[-a2,]

plot(konur.union)

plot(pop.grid.sil, add=TRUE, border="grey80")

points(pts1, pch=".")

pts1.sp<-pts1



# Shannon H for a city

coordinates(pts1.sp)<-c("x", "y")

proj4string(pts1.sp)<-CRS("+proj=merc +datum=NAD83")

pts1.sp<-spTransform(pts1.sp, CRS("+proj=merc +datum=NAD83"))

konur.owin<-as(konur.union, "owin")

konur.ppp<-ppp(x=pts1.sp@coords[,1], y=pts1.sp@coords[,2], window=konur.owin)

konur.tes<-dirichlet(konur.ppp) # Dirichlet tessellation

a<-tile.areas(konur.tes)

a1<-a/sum(a)

ent1<-sum(-1\*a1\*log(a1))

n<-length(a)

ent.ref<-log(1/n)\*(-1)

ent.ref

ent.rel<-ent1/ent.ref # Relative H

ent.rel

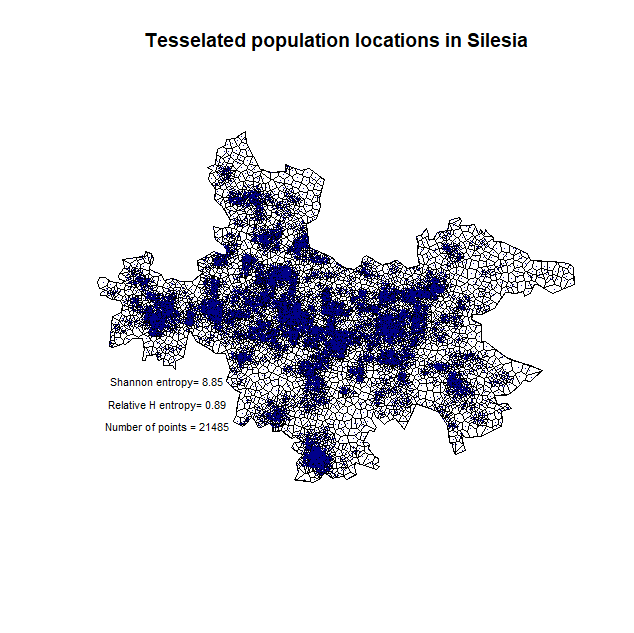
plot(konur.tes, main="Tesselated population locations in Silesia")

plot(konur.ppp, add=TRUE, pch=".", col="darkblue", cex=0.8)

text(2080000, 6450000, paste("Shannon entropy=", round(ent1,2)), cex=0.7)

text(2080000, 6445000, paste("Relative H entropy=", round(ent.rel,2)), cex=0.7)

text(2080000, 6440000, paste("Number of points =", round(n,2)), cex=0.7)



# # Fig.6e – Business locations in Lublin city

firmy<-read.csv("dane\_lubelskie\_FULL.csv", header=TRUE, dec=".", sep=",")

firmy.sp<-firmy

coordinates(firmy.sp)<-c("coords.x1", "coords.x2")

proj4string(firmy.sp)<-CRS("+proj=longlat +datum=NAD83")

firmy.sp<-spTransform(firmy.sp, CRS("+proj=merc +datum=NAD83"))

pow<-readOGR(".", "powiaty") # 16 jedn.

pow<-spTransform(pow, CRS("+proj=merc +datum=NAD83"))

lub<-pow[pow@data$jpt\_nazwa\_=="powiat Lublin",]

plot(lub)

points(firmy.sp, pch=".")

lub.dane<-over(firmy.sp, lub)

a<-which(lub.dane$iip\_przest=="PL.PZGIK.200")

firmy.sp<-firmy.sp[a,]

plot(lub)

points(firmy.sp, pch=".")

lub.owin<-as(lub, "owin")

lub.ppp<-ppp(x=firmy.sp@coords[,1], y=firmy.sp@coords[,2], window=lub.owin)

lub.tes<-dirichlet(lub.ppp) # Dirichlet tessellation

a<-tile.areas(lub.tes)

a1<-a/sum(a)

ent1<-sum(-1\*a1\*log(a1))

n<-length(a)

ent.ref<-log(1/n)\*(-1)

ent.ref

ent.rel<-ent1/ent.ref # Relative H

ent.rel

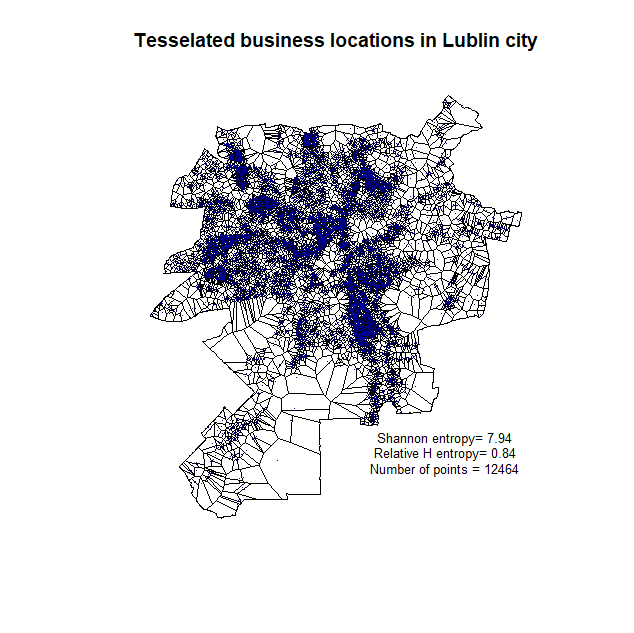
plot(lub.tes, main="Tesselated business locations in Lublin city")

plot(lub.ppp, add=TRUE, pch=".", col="darkblue", cex=0.8)

text(2519000, 6618000, paste("Shannon entropy=", round(ent1,2)), cex=0.8)

text(2519000, 6617000, paste("Relative H entropy=", round(ent.rel,2)), cex=0.8)

text(2519000, 6616000, paste("Number of points =", round(n,2)), cex=0.8)



# # Fig.6f - Analysis of gridded population in Lublin city

lim<-over(pop.grid, lub)

pop.grid.lub<-pop.grid[lim$jpt\_nazwa\_=="powiat Lublin", ]

a<-which(lim$jpt\_nazwa\_=="powiat Lublin")

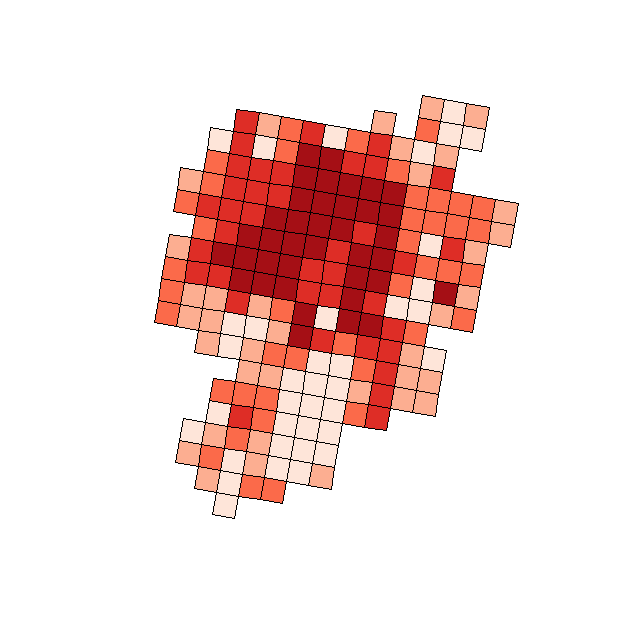
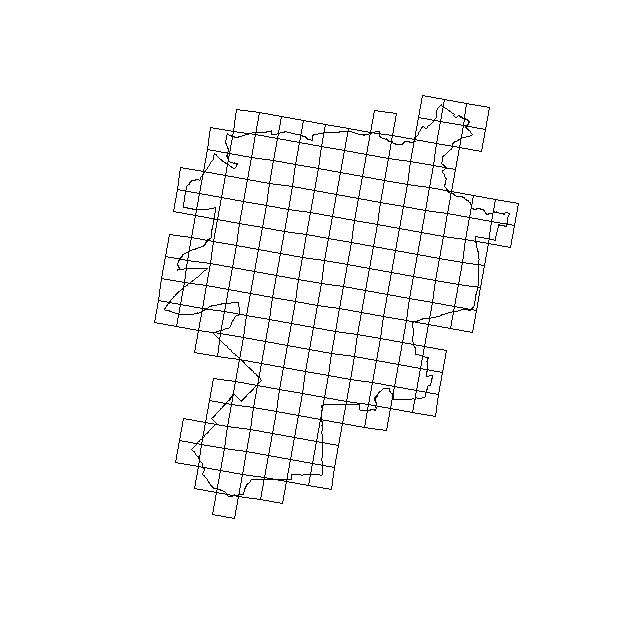
pop.df.lub<-pop.df[a,]

plot(pop.grid.lub)

plot(lub, add=TRUE)

library(GISTools)

choropleth(pop.grid.lub, pop.df.lub$TOT)



**# sampling points within grid cells**

a<-which(pop.df.lub$TOT<100)

pop.df.lub$TOT[a]<-100

pts<-data.frame(x=0, y=0)

for(i in 1:190){

probka<-spsample(pop.grid.lub[i,], pop.df.lub$TOT[i]/100, type="random", iter=10)

probka.df<-as.data.frame(probka@coords)

colnames(probka.df)<-c("x", "y")

pts<-rbind(pts, probka.df)}

pts<-pts[-1,]

plot(lub)

plot(pop.grid.lub, add=TRUE, border="grey80")

points(pts, pch=".")

pts1<-pts

pts.sp<-pts

coordinates(pts.sp)<-c("x", "y")

proj4string(pts.sp)<-CRS("+proj=merc +datum=NAD83")

pts.lim<-over(pts.sp, lub)

a2<-which(is.na(pts.lim)==TRUE)

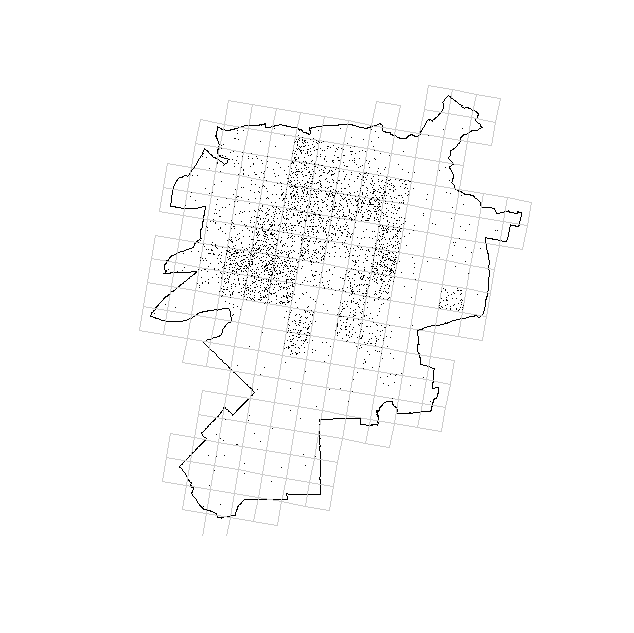
pts1<-pts1[-a2,]

plot(lub)

plot(pop.grid.lub, add=TRUE, border="grey80")

points(pts1, pch=".")

pts1.sp<-pts1



**# Shannon H for city**

coordinates(pts1.sp)<-c("x", "y")

proj4string(pts1.sp)<-CRS("+proj=merc +datum=NAD83")

pts1.sp<-spTransform(pts1.sp, CRS("+proj=merc +datum=NAD83"))

lub.owin<-as(lub, "owin")

lub.ppp<-ppp(x=pts1.sp@coords[,1], y=pts1.sp@coords[,2], window=lub.owin)

lub.tes<-dirichlet(lub.ppp) # Dirichlet tessellation

a<-tile.areas(lub.tes)

a1<-a/sum(a)

ent1<-sum(-1\*a1\*log(a1))

n<-length(a)

ent.ref<-log(1/n)\*(-1)

ent.ref

ent.rel<-ent1/ent.ref # Relative H

ent.rel

plot(lub.tes, main="Tesselated population locations in Lublin city")

plot(lub.ppp, add=TRUE, pch=".", col="darkblue", cex=0.8)

text(2519000, 6618000, paste("Shannon entropy=", round(ent1,2)), cex=0.8)

text(2519000, 6617000, paste("Relative H entropy=", round(ent.rel,2)), cex=0.8)

text(2519000, 6616000, paste("Number of points =", round(n,2)), cex=0.8)

