**R Codes**

**QDC – Quick Density Clustering**

**by Katarzyna Kopczewska**

**Kkopczewska@wne.uw.edu.pl**

**# starter**

library(ggplot2)

library(dbscan)

library(factoextra)

library(clv) # for Rand Index

library(FeatureImpCluster) # for FeatureImpCluster()

library(sp) # for sp projection

library(flexclust) # for kcca conversion

# setting working directory

setwd("C:/my\_folder")

# reading population data

popul.full<-read.csv("points\_popul\_maz.csv", header=TRUE, dec=".", sep=",")

set.seed(123)

selected<-sample(1:65560, 5000, replace=FALSE)

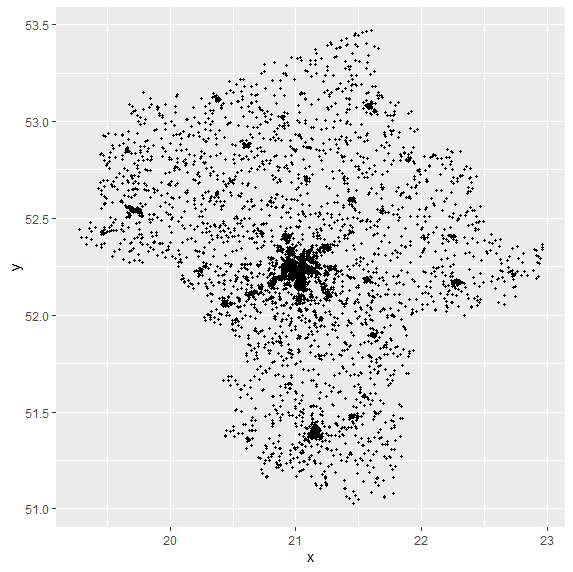
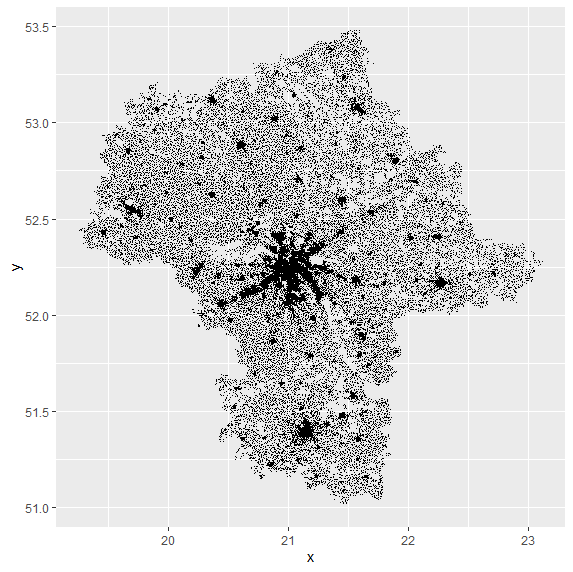
popul<-popul.full[selected, ]

**# Figure 1a - full sample figure**

ggplot(popul.full, aes(x = x, y=y)) + geom\_point(shape=".")

**# Figure 1b - selected observations**

ggplot(popul, aes(x = x, y=y)) + geom\_point(shape=20, size=1)



**# Figure 3a & 3b – DBSCAN with set of parameters**

db<-dbscan(popul, eps=0.05, minPts=25) # select appropriate param.

db<-dbscan(popul, eps=0.15, minPts=50)

popul$db<-db$cluster+1

popul.sp<-popul

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

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

popul.sp<-spTransform(popul.sp, CRS("+proj=longlat +datum=NAD83"))

par(bg="grey93") # colour of background

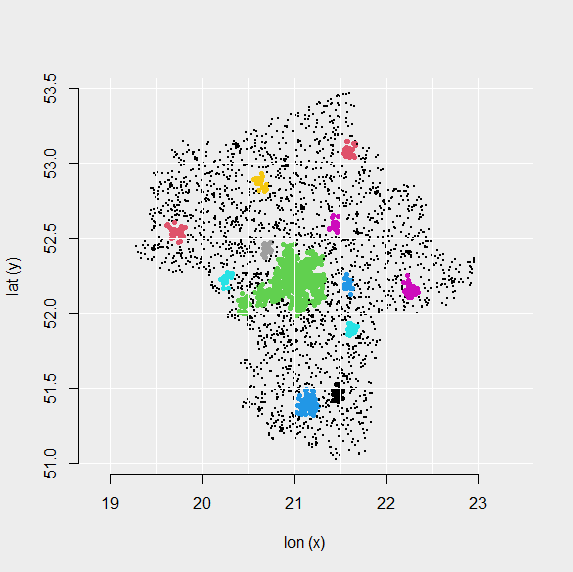
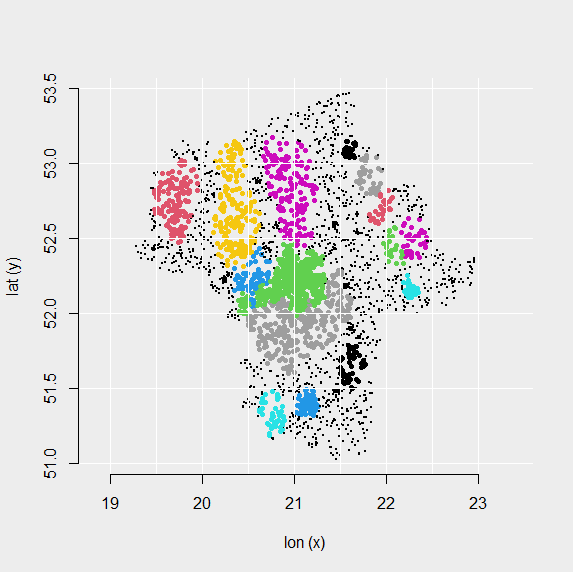
plot(popul.sp, col=popul$db, bg=popul$db, pch=ifelse(db$cluster==0, 20, 21), cex=ifelse(db$cluster==0, 0.05, 0.65), xlab="lon (x)", ylab="lat (y)")

abline(h=c(51, 51.5, 52, 52.5, 53, 53.5), col="white", lwd=1, lty=1)

abline(v=c(19, 19.5, 20, 20.5, 21, 21.5, 22, 22.5, 23), col="white", lwd=1, lty=1)

axis(1)

axis(2)

**# Figure 4 – k-means for geo-coordinates**

km1<-kmeans(popul[,1:2], 5)

popul$km1<-km1$cluster

par(bg="grey93") # colour of background

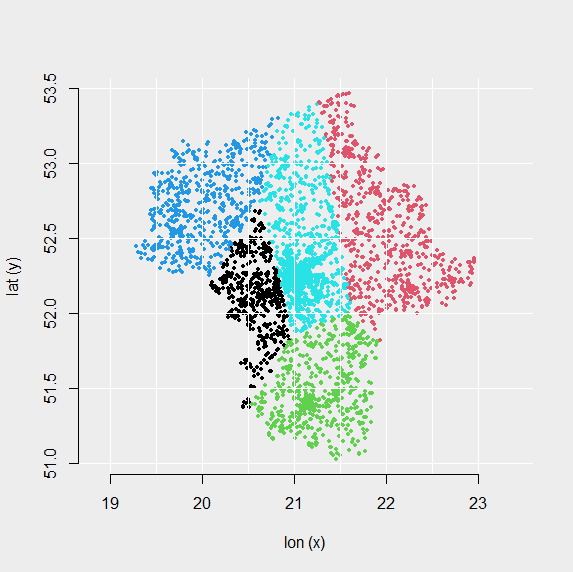
plot(popul.sp, col=km1$cluster, bg=km1$cluster, pch=ifelse(km1$cluster==0, 20, 21), cex=ifelse(km1$cluster==0, 0.05, 0.55), xlab="lon (x)", ylab="lat (y)")

abline(h=c(51, 51.5, 52, 52.5, 53, 53.5), col="white", lwd=1, lty=1)

abline(v=c(19, 19.5, 20, 20.5, 21, 21.5, 22, 22.5, 23), col="white", lwd=1, lty=1)

axis(1)

axis(2)



**# sets of spatial variables**

**# set 1 – knn=10, radius=0.05**

knn.dist<-kNNdist(popul[,1:2], 10, all =TRUE)

popul$knndist1<-apply(knn.dist,1,sum)

agg.radius<-frNN(as.matrix(popul[,1:2]), eps=0.05)

popul$frnn1<-unlist(lapply(agg.radius$id, length))

**# set 2 – knn=30, radius=0.15**

knn.dist<-kNNdist(popul[,1:2], 30, all =TRUE)

popul$knndist2<-apply(knn.dist, 1, sum)

agg.radius<-frNN(as.matrix(popul[,1:2]), eps=0.15)

popul$frnn2<-unlist(lapply(agg.radius$id, length))

**# normalisation of variables**

popul$x.scaled<-scale(popul$x)

popul$y.scaled<-scale(popul$y)

popul$knndist1.scaled<-scale(popul$knndist1)

popul$frnn1.scaled<-scale(popul$frnn1)

popul$knndist2.scaled<-scale(popul$knndist2)

popul$frnn2.scaled<-scale(popul$frnn2)

**# density distributions of spatial variables**

**# Fig.6**

par(bg="grey93") # colour of background

par(mar=c(3,3,3,3))

plot(density(popul$knndist1.scaled), ylim=c(0,3), lwd=2, main="Density distribution of spatial variables")

lines(density(popul$frnn1.scaled))

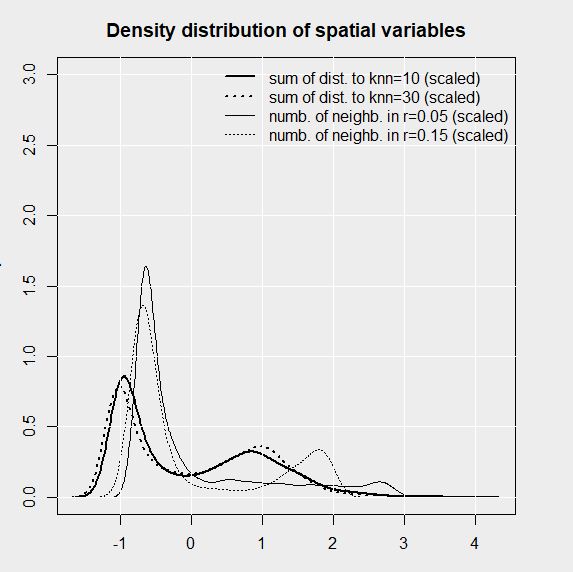
lines(density(popul$knndist2.scaled), lwd=2, lty=3)

lines(density(popul$frnn2.scaled), lty=3)

abline(h=(0:6)\*0.5, col="white", lwd=1, lty=1)

abline(v=(-2:10)\*1, col="white", lwd=1, lty=1)

legend("topright", legend=c("sum of dist. to knn=10 (scaled)", "sum of dist. to knn=30 (scaled)", "numb. of neighb. in r=0.05 (scaled)", "numb. of neighb. in r=0.15 (scaled)"), lty=c(1,3,1,3), lwd=c(2,2,1,1), bty="n")



**# clustering of spatial variables – set 1**

**# Fig.7**

fviz\_nbclust(popul[,11:12], FUNcluster=kmeans) # factoextra::

km.set1<-kmeans(popul[ ,11:12], 3)

popul$km.set1<-as.factor(km.set1$cluster)

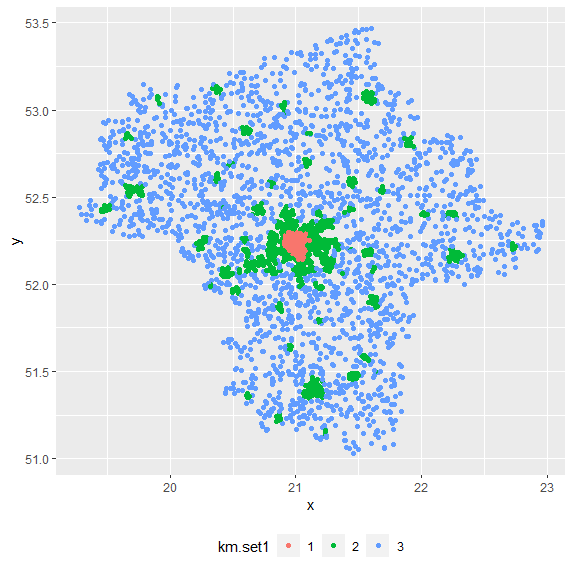
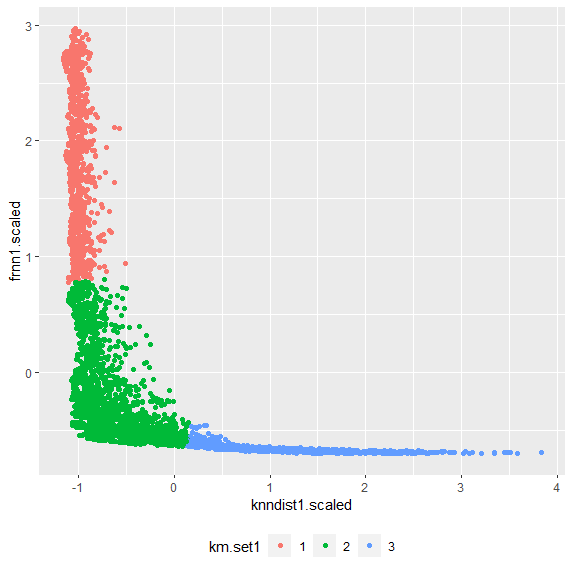
ggplot(popul, aes(x=knndist1.scaled, y=frnn1.scaled, color=km.set1)) + geom\_point()+ theme(legend.position="none")

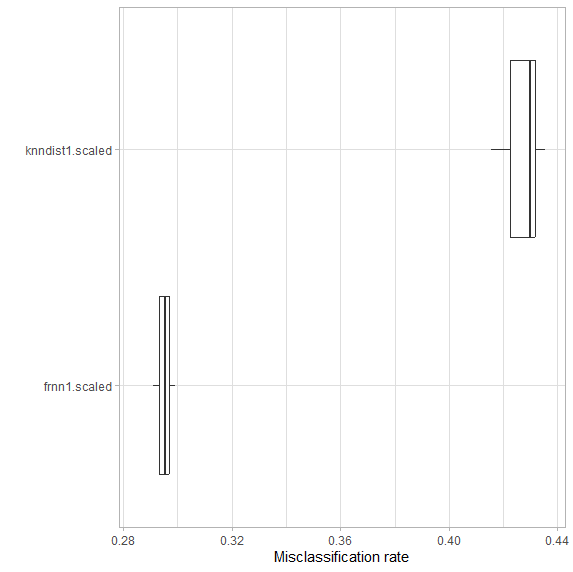
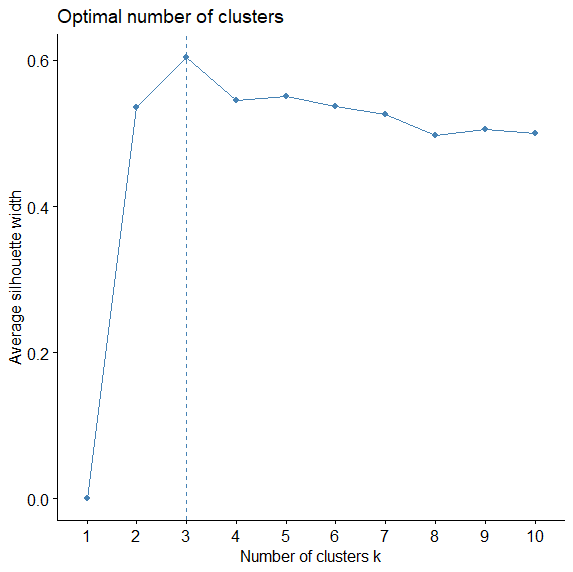
ggplot(popul, aes(x=x, y=y, color=km.set1)) + geom\_point() + theme(legend.position="none") # for legend "bottom"

clust\_kmeans<-as.kcca(km.set1, popul[,11:12]) # conversion to kcca

FeatureImp\_km<-FeatureImpCluster(clust\_kmeans, as.data.table(popul[,11:12]))

plot(FeatureImp\_km)





**# clustering of spatial variables – set 2**

**# Fig.8**

fviz\_nbclust(popul[,13:14], FUNcluster=kmeans) # factoextra

km.set2<-kmeans(popul[ ,12:13], 3)

popul$km.set2<-as.factor(km.set2$cluster)

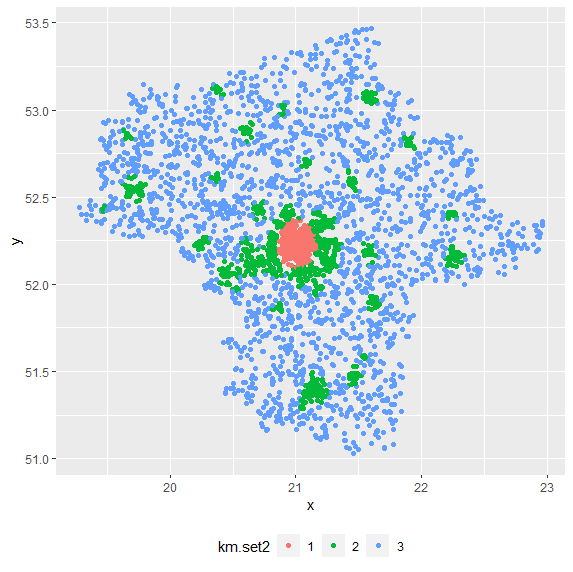
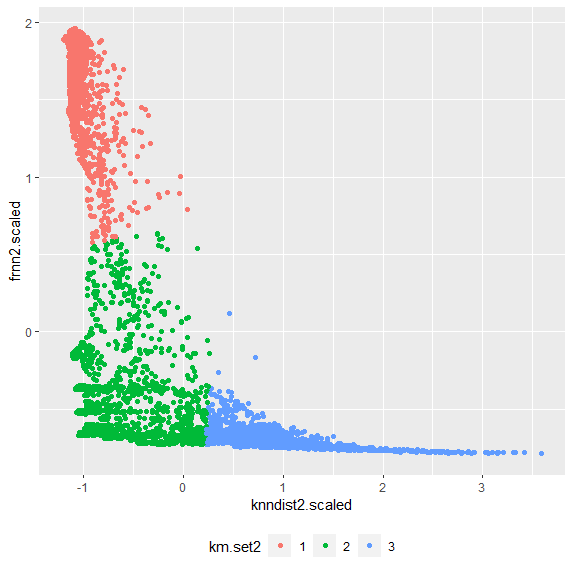
ggplot(popul, aes(x=knndist2.scaled, y=frnn2.scaled, color=km.set2)) + geom\_point()+ theme(legend.position="none")

ggplot(popul, aes(x=x, y=y, color=km.set2)) + geom\_point()+ theme(legend.position="none")

clust\_kmeans<-as.kcca(km.set2, popul[,12:13]) # conversion to kcca

FeatureImp\_km<-FeatureImpCluster(clust\_kmeans, as.data.table(popul[,12:13]))

plot(FeatureImp\_km)



**# Rand index – to compare both partitioning**

external.ind<-std.ext(as.vector(popul$km.set1), as.vector(popul$km.set2))

clv.Rand(external.ind)

# 0.9255842

**# thresholds for cluster assignement**

t1<-max(min(popul$knndist1.scaled[popul$km.set1==1]),

min(popul$knndist1.scaled[popul$km.set1==2]),

min(popul$knndist1.scaled[popul$km.set1==3]))

t1 # when knndist>t1 – it is low-density cluster

t2<-max(min(popul$frnn1.scaled[popul$km.set1==1]),

min(popul$frnn1.scaled[popul$km.set1==2]),

min(popul$frnn1.scaled[popul$km.set1==3]))

t2 # when frnn(agg)>t2 – it is high-density cluster

popul$outcome.set1<-ifelse(popul$knndist1.scaled>t1, "low-density", ifelse(popul$frnn1.scaled>t2,"high-density", "mid-density"))

**#QDC - full clustering procedure**

head(popul) # dataset with geo-coordinates only

x y

51973 21.28148 51.79619

58212 21.64510 51.90861

3105 19.72021 52.54695

30185 21.02671 52.29732

63048 22.30499 52.72241

45686 21.23054 52.21566

knn.dist<-kNNdist(popul[,1:2], 10, all =TRUE) # spatial variables

popul$knndist1<-apply(knn.dist,1,sum) # sum of distances to knn

agg.radius<-frNN(as.matrix(popul[,1:2]), eps=0.05) # neighb. in r

popul$frnn1<-unlist(lapply(agg.radius$id, length)) # count

popul$knndist1.scaled<-scale(popul$knndist1) # normalisation

popul$frnn1.scaled<-scale(popul$frnn1)

km.set1<-kmeans(popul[ ,5:6], 3) # kmeans clustering

popul$km.set1<-as.factor(km.set1$cluster)

t1<-max(min(popul$knndist1.scaled[popul$km.set1==1]),

min(popul$knndist1.scaled[popul$km.set1==2]),

min(popul$knndist1.scaled[popul$km.set1==3])) # threshold

t1 # when knndist>t1 – it is low-density cluster

t2<-max(min(popul$frnn1.scaled[popul$km.set1==1]),

min(popul$frnn1.scaled[popul$km.set1==2]),

min(popul$frnn1.scaled[popul$km.set1==3])) # threshold

t2 # when frnn(agg)>t2 – it is high-density cluster

# classification of points to clusters

popul$outcome.set1<-ifelse(popul$knndist1.scaled>t1, "low-density", ifelse(popul$frnn1.scaled>t2,"high-density", "mid-density"))

head(popul)

x y knndist1 frnn1 knndist1.scaled frnn1.scaled km.set1 outcome.set1

51973 21.28148 51.79619 0.66723922 2 1.5592872 -0.68661635 1 low-density

58212 21.64510 51.90861 0.25665720 23 -0.1550395 -0.50108352 3 mid-density

3105 19.72021 52.54695 0.05244444 83 -1.0077007 0.02901027 3 mid-density

30185 21.02671 52.29732 0.06815969 177 -0.9420839 0.85949053 2 high-density

63048 22.30499 52.72241 0.47036912 5 0.7372842 -0.66011166 1 low-density

45686 21.23054 52.21566 0.18227659 21 -0.4656051 -0.51875331 3 mid-density

ggplot(popul, aes(x=knndist1.scaled, y=frnn1.scaled, color=km.set1)) + geom\_point()+ theme(legend.position="none") # xy plot of spat.var

ggplot(popul, aes(x=x, y=y, color=km.set1)) + geom\_point()+ theme(legend.position="none") # location of clusters

