Appendix: R code

rm(list = ls())

setwd("C:/Users/tuj53509/Dropbox/docs/Temple/Advanced Statistics for Urban Applications/Assignment3")

library(sp)

library(rgdal)

library(maptools)

library(RColorBrewer)

library(GISTools)

require(SpatialEpi)

require(deldir)

library(gstat)

library(classInt)

#load data

midatl\_states <- readShapePoly("MidAtlStates\_WGS/MidAtlStates\_WGS.shp", proj4string = CRS("+init=EPSG:4326"))

names(midatl\_states)

plot(midatl\_states)

greenhouse <- read.csv("2015EPA\_FLIGHT\_GreenhouseGas.csv", header = TRUE)

greenhouse <- as.data.frame(greenhouse)

head(greenhouse)

ghg <- greenhouse[,c("x","y")]

head(ghg)

ghg.spdf <- SpatialPointsDataFrame(greenhouse[,c("x","y")], data = greenhouse, proj4string = CRS("+init=EPSG:4326"))

head(ghg.spdf)

par(mfrow=c(1,1))

par(mar=c(0,0,0,0))

plot(ghg.spdf)

ghg.utm <- spTransform(ghg.spdf, CRS("+init=EPSG:32618"))

sts.utm <- spTransform(midatl\_states, CRS("+init=EPSG:32618"))

pdf(file="f1.ghgandstates.pdf")

par(mar=c(2,2,2,2))

plot(ghg.utm, main="Figure 1: Midatlantic Carbon Equivalent of Emissions")

plot(sts.utm, add = TRUE)

dev.off()

pdf(file="f2.ghghist.pdf")

par(mfrow=c(1,1), mar=c(2,2,2,2))

hist(ghg.utm$CarbonMetTon, breaks = 10, main="Figure 2: Distribution of Carbon Equivalent of Emissions, 2015")

dev.off()

#Step 3: Nearest Neighbor Estimation

#set up functions

voronoipolygons = function(layer) {

crds <- layer@coords

z <- deldir(crds[,1], crds[,2])

w <- tile.list(z)

polys <- vector(mode='list', length=length(w))

for (i in seq(along=polys)) {

pcrds <- cbind(w[[i]]$x, w[[i]]$y)

pcrds <- rbind(pcrds, pcrds[1,])

polys[[i]] <- Polygons(list(Polygon(pcrds)),

ID=as.character(i))

}

SP <- SpatialPolygons(polys)

voronoi <- SpatialPolygonsDataFrame(SP,

data=data.frame(dummy=sapply(slot(SP, 'polygons'),

function(x) slot(x, 'ID'))))

return(voronoi)

}

ghg.voro <- voronoipolygons(ghg.utm)

pdf(file="f3.nearestneighbor.pdf")

par(mfrow=c(1,2),mar=c(1,1,4,1))

plot(ghg.spdf)

title("Figure 3: Nearest Neighbor Diagnostic")

plot(ghg.voro)

dev.off()

par(mfrow=c(1,1), mar=c(2,0,2,0))

pdf(file="f4.nearestneighborchoro.pdf")

nneigh.shades <- auto.shading(ghg.utm$CarbonMetTon, cols=brewer.pal(5,"Blues"))

choropleth(ghg.voro, ghg.utm$CarbonMetTon, shading=nneigh.shades, main="Figure 4: Carbon Equivalent of Emissions, 2015")

choro.legend(px='bottomright',bg="white",sh=nneigh.shades, cex=.75, title="Metric Tons of Greenhouse Gases")

dev.off()

#create two KDEs

par(mfrow=c(1,1))

#KDE1 - use the default bandwidth

ghg1.dens <- kde.points(ghg.utm, lims = sts.utm)

masker1 <- poly.outer(ghg1.dens, sts.utm, extend = 100)

pdf(file="f5.kde01.pdf")

level.plot(ghg1.dens)

add.masking(masker1)

plot(sts.utm, add = TRUE)

title("Figure 5: Density Estimate of Carbon Emissions")

dev.off()

#KDE2 - use manually set a bandwidth

ghg2.dens <- kde.points(ghg.utm, h=53000, lims = sts.utm)

masker2 <- poly.outer(ghg2.dens, sts.utm, extend = 100)

pdf(file="f6.kde02.pdf")

level.plot(ghg2.dens)

add.masking(masker2)

plot(sts.utm, add = TRUE)

title("Figure 6: Density Estimate of Carbon Emissions, increased bandwidth", cex = .75)

dev.off()

#Step 4: Inverse Distance Weighting

proj4string(ghg.voro) <- CRS("+init=EPSG:32618")

ghg.grid1 <- spsample(ghg.voro,type='regular',n=6000)

#model 1 with alpha = 1.0

ghg.idw.est1 <- gstat::idw(CarbonMetTon~1, ghg.utm,newdata=ghg.grid1,idp=1.0)

#Model 2 with alpha = 2.0

ghg.idw.est2 <- gstat::idw(CarbonMetTon~1, ghg.utm,newdata=ghg.grid1,idp=2.0)

ux <- unique(coordinates(ghg.idw.est1)[,1])

uy <- unique(coordinates(ghg.idw.est1)[,2])

#prediction matrix for idw estimate 1

predmat <- matrix(ghg.idw.est1$var1.pred,length(ux),length(uy))

#prediction matrix for idw estimate 2

predmat2 <- matrix(ghg.idw.est2$var1.pred,length(ux),length(uy))

#calculate breaks

#classbreaks <- classIntervals(ghg.idw.est1$var1.pred,n=5,style="jenks")

#plot idw estimates

pdf(file="f7.idwests.pdf")

par(mar=c(1,1,2,0.1),mfrow=c(1,2))

plot(ghg.voro,border=NA,col=NA)

.filled.contour(ux,uy,predmat,col=brewer.pal(6,'Blues'),

levels=c(0,125672,308036,352758,434444,682066,1385206))

title("Figure 7: IDW estimate, alpha=1", cex=0.75)

#levels=c(125672,308036,352758,434444,682066,1385206))

#plot IDW estimate 2

plot(ghg.voro,border=NA,col=NA)

.filled.contour(ux,uy,predmat2,col=brewer.pal(6,'Blues'),

levels=c(0,125672,308036,352758,434444,682066,1385206))

title("alpha=2", cex=0.75)

dev.off()

#Step 5: Kriging

#,boundaries=seq(0,500000,l=100)

#ghg.vari.fit <- fit.variogram(ghg.vari.est,vgm(1,"Mat",1200000,l=51))

#ghg.vari.est <- variogram(CarbonMetTon~1,ghg.utm)

ghg.vari.est <- variogram(CarbonMetTon~1,ghg.utm)

plot(ghg.vari.est)

#ghg.vari.fit <- fit.variogram(ghg.vari.est,vgm(1, "Mat", 1290000, 1))

ghg.vari.fit <- fit.variogram(ghg.vari.est,vgm("Mat"))

plot(ghg.vari.est)

#try using eyefit package to automatically get a starting fit

install.packages("geoR")

library(geoR)

eyefit(ghg.vari.est)

plot(ghg.vari.est,model=ghg.vari.fit)