Kernel Density Ratios and Network Kernel Densities

Yalin Yang

2020-02-27

# Task 1: Supply and Demand Imbalances [11 points]

In this task you will be working predominately with the layers **blockGroupShp, tractShp, bndShp** and **foodStoresShp** of the package **TexMix**. Additional information can be found in Baddely (see **RSPATIALCOURSE\_CMIS\_PDF** **STANDARD.PDF**)

### helper packages  
library(TexMix); library(spatstat); library(smacpod); library(colorspace)  
library(sp); library(maptools); library(rgdal)  
  
### loading data  
data("foodStoresShp"); data("blockgroupShp")

### 1.01: Re-project the layers into the UTM system [0.5 points]

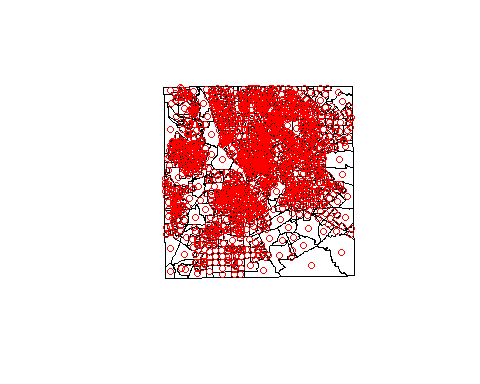
## Project to UTM  
(projOld <- proj4string(bndShp))

## [1] "+proj=longlat +ellps=GRS80 +no\_defs"

projUTM <- CRS("+proj=utm +zone=14 +units=m") # New coordinate sytem definition  
bndUTM <- spTransform(bndShp, projUTM) # Re-project boundary  
stUTM <- spTransform(foodStoresShp, projUTM) # Re-project store locations  
tractUTM <- spTransform(tractShp, projUTM) # Re-project census tracts  
bgUTM <- spTransform(blockgroupShp, projUTM) # Re-project block groups

### 1.02: Extract the center of gravity form the block groups [0.5 points]

pts <- coordinates(bgUTM)  
plot(bgUTM)  
points(pts,col = 'red')



### 1.03: Set a window mask for the study area with a pixel resolution of 200 meters [0. 5 points]

win <- as.mask(as.owin(bndUTM), eps=200) # Pixel mask with 200 m resolution

### 1.04: Save the block group nighttime population as well as the food sales of the food stores as weights. [0. 5 points]

bgWgh <- blockgroupShp$NIGHTPOP  
stWgh <- foodStoresShp$FOODSALES

### 1.05: Convert the block group centroids and the food store locations into ppp-objects and remove all marks. Note see p 51 on creating ppp’s in the Baddely document [0.5 points]

bgPts <- ppp(pts[,1], pts[,2], window=win)  
bgPts$marks <- NULL  
  
stPts <- as.ppp(stUTM)  
stPts$marks <- NULL  
stPts <- stPts[win]

### 1.06: Convert the census tracts boundaries into tessellation windows as well as a tessellation image with a 200-meter pixel resolution. [0.5 points]

## Convert tracts to tesselation object for mapping  
tracts <- slot(tractUTM, "polygons")  
tracts <- lapply(tracts, function(x) { SpatialPolygons(list(x)) })  
tracts <- lapply(tracts, as.owin)  
tractsTess <- tess(tiles=tracts)  
tractsMask <- as.mask(tractsTess, eps=200)

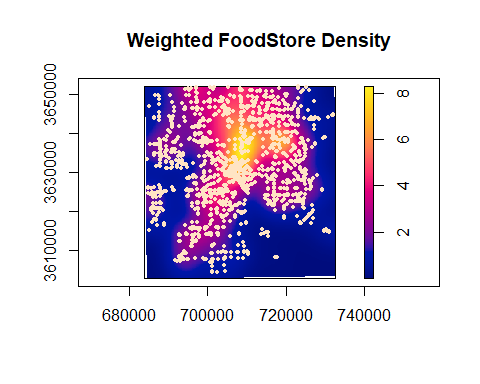
### 1.07: Select for the sales volume weighted kernel density of the food stores the appropriate bandwidth. Use sustentative arguments for your bandwidth choice. Map the resulting density with the food store locations and census tract boundaries superimposed. This density will become your food supply pattern. [1.5 points]

bw.scott(stPts) # scott's (x,y)-direction evaluation

## sigma.x sigma.y   
## 3272.003 3260.975

**So in here, we could choose 3200 as our sigma input.**

plot(density(stPts, sigma = 3200, weights=stWgh), main="Weighted FoodStore Density")  
plot(stPts, cex=0.5, pch=16, col="bisque", add=T)  
box(); axis(1); axis(2)



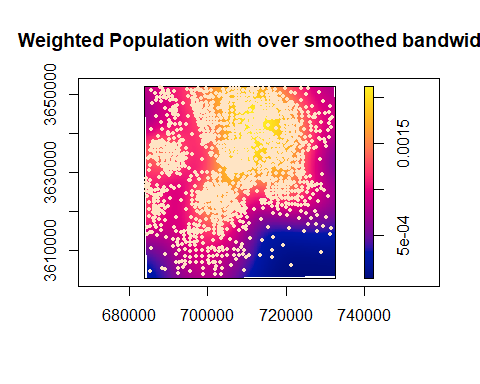
### 1.08: Find for the population weighted block group centroids the optimal smoothing bandwidth. Justify your choice by comparing this bandwidth with an over- and an under-smoothed pattern. Map the resulting population density with the census tract boundaries superimposed. This density will become your food demand pattern. [1.5 points]

bw.scott(bgPts) # scott's (x,y)-direction evaluation

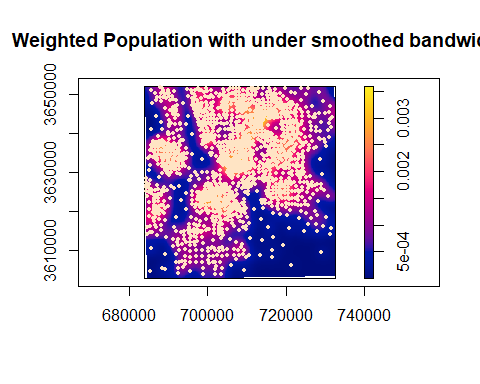
## sigma.x sigma.y   
## 3294.696 3284.166

**So the proper bandwidth would also be 3200.**

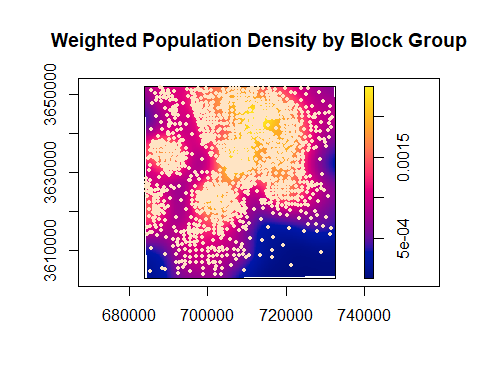
plot(density(bgPts, sigma= 4500, weights=bgWgh), main="Weighted Population with over smoothed bandwidth")  
plot(bgPts, cex=0.5, pch=16, col="bisque", add=T)  
box(); axis(1); axis(2)



plot(density(bgPts, sigma= 1500, weights=bgWgh), main="Weighted Population with under smoothed bandwidth")  
plot(bgPts, cex=0.5, pch=16, col="bisque", add=T)  
box(); axis(1); axis(2)

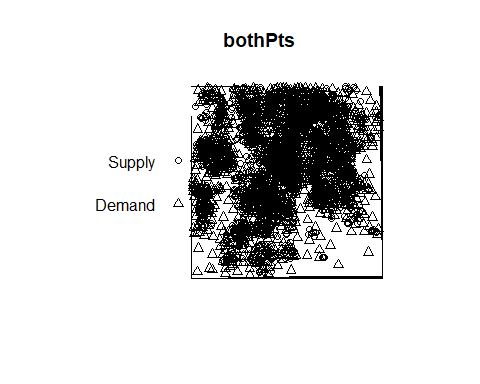


plot(density(bgPts, sigma=3200, weights=bgWgh), main="Weighted Population Density by Block Group")  
plot(bgPts, cex=0.5, pch=16, col="bisque", add=T)  
box(); axis(1); axis(2)



### 1.09: Merge the food store locations and block group population centroids as marked point pattern into one ppp-object. See Baddely p 61. Also combine both weights vector into a joint vector. Make sure that food stores are at the top of the merged ppp-object and the combined weights vector. [1 point]

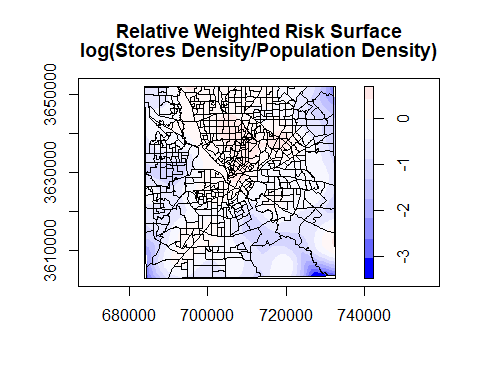
## Merge both points and assign marks  
bothPts <- superimpose.ppp(Supply=stPts, Demand=bgPts, W=win)  
plot(bothPts)



weights <-c(stWgh,bgWgh)

### 1.10: Calculate and properly map the log relative risk, i.e., log(supply⁄demand), with the previously identified bandwidths and weights vector. Interpret the resulting map pattern. [1 point]

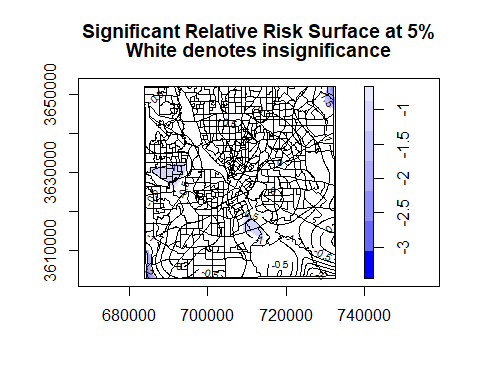
## Relative Risk Evaluation  
riskMap <- logrr(bothPts, case=1, sigma=3000, sigmacon=3000, edge=FALSE, weights=c(stWgh,bgWgh))  
bound <- max(abs(range(riskMap$v, na.rm=TRUE)))  
plot(riskMap, main="Relative Weighted Risk Surface\nlog(Stores Density/Population Density)",  
 breaks=seq(-bound, bound, length.out=16+1), col=diverge\_hsv(16))  
plot(tractsTess, col="grey", add=T)  
box(); axis(1); axis(2)



**From the map above, we notice that the central and north areas are slightly over-supplied. A big proportion of the boundary areas is under-supplied, which may caused by spatial uncertainty.**

### 1.11: Evaluate the two-sided significant of the over- and under-supplied areas at a significance level of α=0.05 and discuss the resulting map. [1 point]

## Significance Evaluation  
riskMapSig <- logrr(bothPts, case=1, level=0.95, alternative="two.sided",  
 sigma=3000, sigmacon=3000, nsim=99, weights=c(stWgh,bgWgh))  
bound <- 3.5  
plot(riskMapSig,  
 main="Significant Relative Risk Surface at 5%\nWhite denotes insignificance",  
 breaks=seq(-bound, bound, length.out=16+1), col=diverge\_hsv(16))  
plot(tractsTess, add=T)  
box(); axis(1); axis(2)



### 1.12: What could be done with regards to the study design to overcome the problem of having apparently significant under-supply at the boundary of Dallas County, i.e., the edges of the study area? [1 point]

**using guard area (the area outside of the boundary of our study area) and also collecting data from those areas. For an island or someplace with a natural barrier from outside, do not need to consider this.**

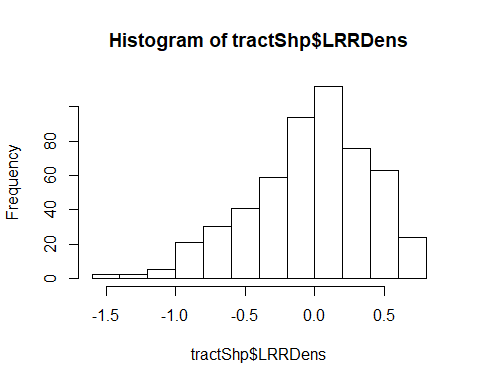
### 1.13: Aggregated the density pixel values into the census tracts. Generate an informative map in the vector domain showing the aggregated densities. Answer the question which designated food deserts in fact suffer from a lack of food supply. [1 point]

densityKernelFct <- function(kernel, tess){  
 ##  
 ## Split a kernel density image into individual tessellations (also type image)  
 ## and calculated for each tessellation descriptive statistics of the kernel  
 ## density pixel values  
 ##  
 tract <- split(kernel, tess) # see Baddeley p 122  
 #sapply(tract, integral.im) # kernel sum in tract  
 n <- length(tract) # number of tracts  
 ## Initialized data-frame of results  
 df <- data.frame(SeqId=1:n, nOfCells=NA, Sum=NA, Density=NA, varDens=NA)  
 ## Cycle over tracts. Data values are in $v and may include NAs.  
 ## $v is a matrix.  
 for (i in 1:n){  
 Sum <- sum(tract[[i]]$v, na.rm = T)  
 Cells <- length(na.omit(as.vector(tract[[i]]$v)))  
 Density <- mean(as.vector(tract[[i]]$v), na.rm = T)  
 varDens <- var(as.vector(tract[[i]]$v), na.rm = T)  
 df[i,2:5] <- c(Cells, Sum, Density, varDens)  
 }  
 return(df)  
} ## end::densityKernelFct

## Evaluate risk by census tract  
tractsTessIm <- tess(tiles=tracts, window=win)  
riskStats <- densityKernelFct(riskMap, tractsTessIm)  
summary(riskStats)

## SeqId nOfCells Sum Density   
## Min. : 1 Min. : 7.0 Min. :-3266.133 Min. :-1.46441   
## 1st Qu.:133 1st Qu.: 37.0 1st Qu.: -21.423 1st Qu.:-0.28075   
## Median :265 Median : 64.0 Median : 1.133 Median : 0.02247   
## Mean :265 Mean : 111.6 Mean : -33.350 Mean :-0.03285   
## 3rd Qu.:397 3rd Qu.: 113.0 3rd Qu.: 11.274 3rd Qu.: 0.26157   
## Max. :529 Max. :3405.0 Max. : 96.803 Max. : 0.69908   
## varDens   
## Min. :0.0000023   
## 1st Qu.:0.0004513   
## Median :0.0017958   
## Mean :0.0095853   
## 3rd Qu.:0.0047895   
## Max. :0.5773943

tractShp$LRRDens <- riskStats$Density  
tractShp$LRRVar <- riskStats$varDens  
hist(tractShp$LRRDens)



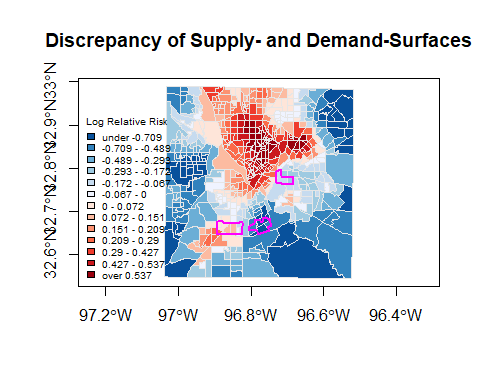
sum(tractShp$LRRDens < 0)

## [1] 254

sum(tractShp$LRRDens >= 0)

## [1] 275

mapBiPolar(tractShp$LRRDens, tractShp, neg.breaks=6, pos.breaks=7, legend.cex = 0.6,  
 map.title="Discrepancy of Supply- and Demand-Surfaces",   
 legend.title= "Log Relative Risk", add.to.map=F)  
plot(foodDesertShp, border="magenta",lwd=2, add=T)



**Combined with the result of the Significance Evaluation map and the consideration of edge effects, we take a 5% risk to say that the central south area is suffered from food shortage.**

# Task 2: Network Kernel Density [3 points]

Use the **StreetNetwork** and **Accidents** layers in the file StreetAccidentsShp.zip. Note: this is an artificial dataset.

rm(list=ls())  
library(spatstat); library(maptools); library(sp)  
  
## Read network shape-file  
streets <- rgdal::readOGR(dsn="./StreetAccidentsShp",   
 layer="StreetNetwork", integer64="warn.loss")

## OGR data source with driver: ESRI Shapefile   
## Source: "C:\Users\Administrator\Downloads\StreetAccidentsShp", layer: "StreetNetwork"  
## with 146 features  
## It has 3 fields  
## Integer64 fields read as signed 32-bit integers: ID

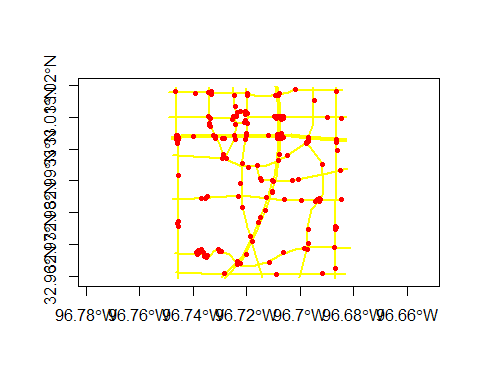
proj4string(streets) # Current map projection

## [1] "+proj=longlat +ellps=GRS80 +no\_defs"

plot(streets, col="yellow", lwd=2, axes=T)  
accidents <- rgdal::readOGR(dsn="StreetAccidentsShp",   
 layer="Accidents", integer64="warn.loss")

## OGR data source with driver: ESRI Shapefile   
## Source: "C:\Users\Administrator\Downloads\StreetAccidentsShp", layer: "Accidents"  
## with 156 features  
## It has 1 fields  
## Integer64 fields read as signed 32-bit integers: ID

plot(accidents, pch=20, col="red", add=T)



### 2.01: Import both shape-files into R code Convert the projection to UTM coordinates. [0.5 points]

## Convert to UTM projections  
UTMDallas <- CRS("+proj=utm +zone=14 +units=m") # New coordinate sytem definition   
streets <- spTransform(streets, UTMDallas) # Transform boundary  
proj4string(streets) # New UTM map projection

## [1] "+proj=utm +zone=14 +units=m +ellps=WGS84"

accidents <- spTransform(accidents, UTMDallas)

### 2.02 Convert street network into a linnet object and the accident locations into a ppp-object. Join the accident locations with the street network into a lpp object. Map the lpp object with green streets and red accident locations. [0.5 points]

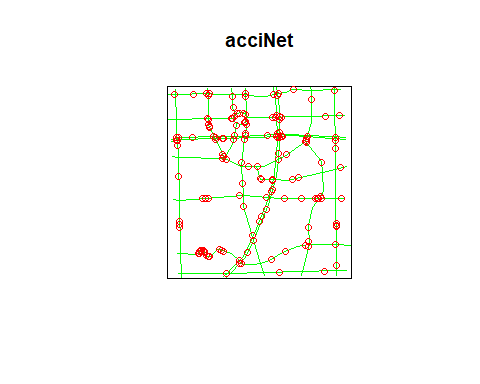
## Convert streets to linnet  
streetsNet <- as.linnet(streets, sparse=F) # Also computes shortest path distances  
summary(streetsNet)

## Linear network with 174 vertices and 221 lines  
## Total length 98865.93 units  
## Maximum vertex degree: 4  
## [Sparse matrix representation]  
## Network is connected  
## Numerical tolerance: 0.0185501 units  
## Enclosing window: rectangle = [710213.7, 716705.5] x [3649042, 3655802]   
## units

win <- as.mask(boundingbox(streetsNet), eps=200)  
streetsNet <- streetsNet[win]

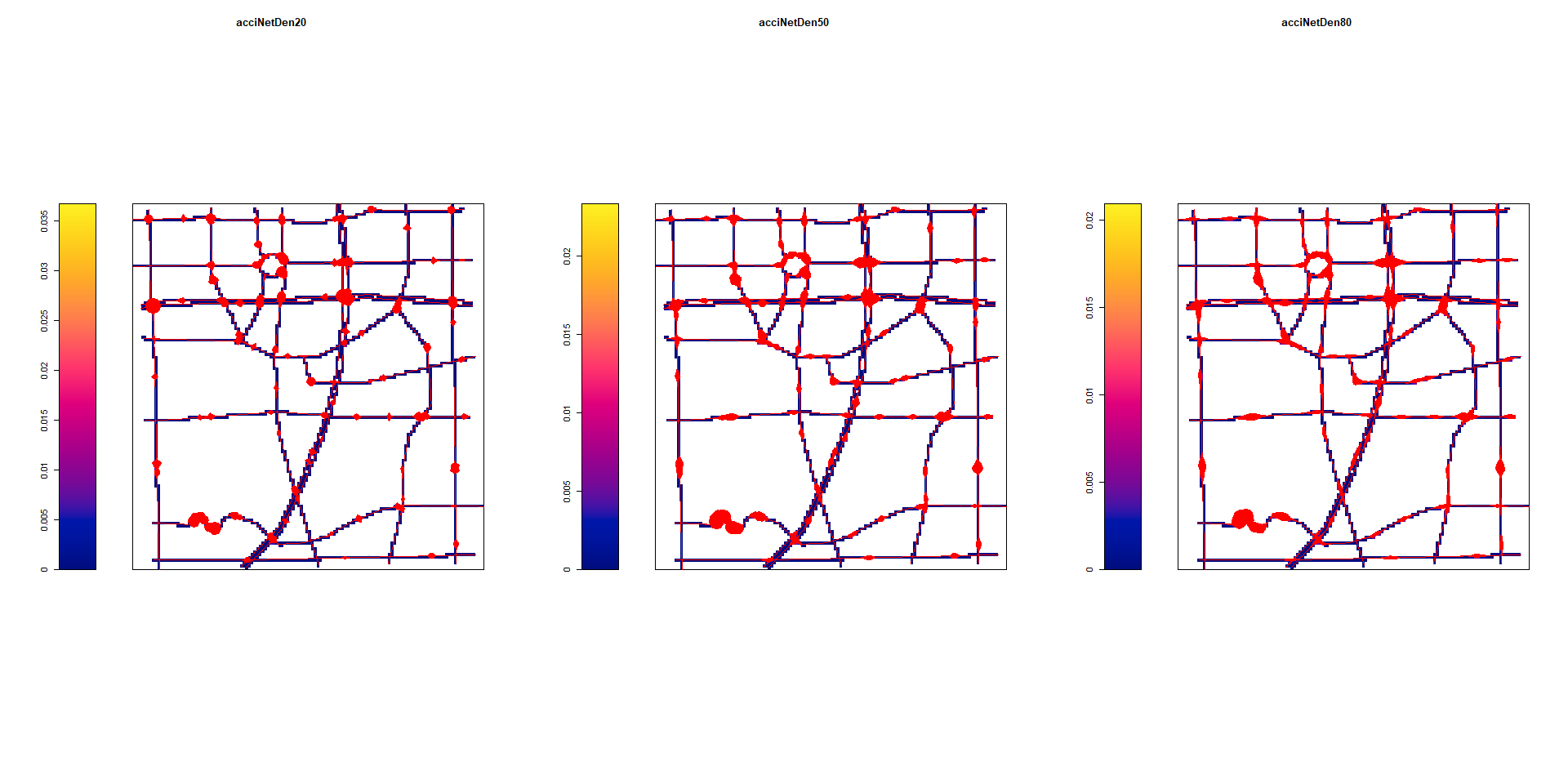
## Convert accidents to ppp  
accidentsPts <- as.ppp(accidents)  
accidentsPts$marks <- NULL   
accidentsPts <- accidentsPts[win]

## Join network and points. Also snaps points onto the network  
acciNet <- lpp(accidentsPts, streetsNet)  
unitname(acciNet) <- c("metre", "metres")  
plot(acciNet, col="green", cols="red")



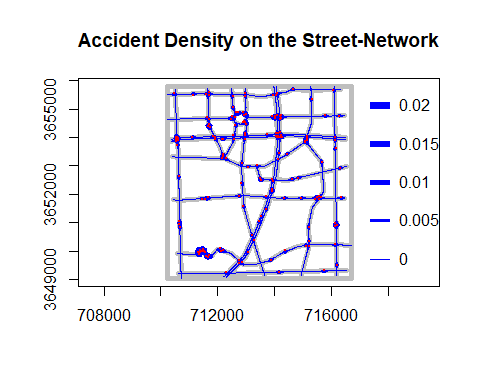
### 2.03 Find and justify an appropriate band-width for the accidents along the street network. Show the results both as a color and a width maps. [0.5 points]

## Inhomogeneous intensity with kernel density  
par(mfrow=c(1,3))  
acciNetDen20 <- density(acciNet, sigma = 20, continuous=T)  
plot(acciNetDen20, style="colour", ribside="left")  
plot(acciNetDen20, style="width", adjust=3, col="red", add=T)  
acciNetDen50 <- density(acciNet, sigma = 50, continuous=T)  
plot(acciNetDen50, style="colour", ribside="left")  
plot(acciNetDen50, style="width", adjust=3, col="red", add=T)  
acciNetDen80 <- density(acciNet, sigma = 80, continuous=T)  
plot(acciNetDen80, style="colour", ribside="left")  
plot(acciNetDen80, style="width", adjust=3, col="red", add=T)



**bandwidth with 30 are too more details, and for avoiding over soomth, select bandwidth with 50 here.**

## Inhomogeneous intensity with kernel density  
acciNetDen <- density(acciNet, sigma = 50, eps=20, continuous=T)  
plot(streetsNet, window=FALSE, col="grey", axes=T, main="",lwd=5)  
plot(acciNetDen, style="width", adjust=3, col="blue", add=T)  
plot(accidentsPts, col="red", pch=20, cex=0.5, add=T )  
title(main="Accident Density on the Street-Network")



## 2.04: Interpret the density of accident locations relative to properties of the street network. [0.5 points]

**From the map above, we could discover that almost all accidents occur in intersections among roads, that’s why from the density map, there are a lot of clusters on there. Besides, roundabouts are also be considered as an area with high frequncy of accidents**

## 2.05: Do median separated streets coded as separate lanes, each having their specific driving direction, have the potential to cause problems in an analysis? [0.5 point]

**No, we need to put the direction of roads into consider. when we drive, the accident happened in another direction definitely was not that much influence on our road.**

## 2.06: Could the analysis be enhanced by accounting for traffic density of each road segment? How can this be implemented in the analysis? [0.5 points]

**Yes, it is helpful for estimating risk of accidents.We could normalize our data using the count of traffic density**