

2015 SISMID Spatial Short Course Pennsylvania Example Solution

On the class website you will find breast cancer incidence data from counties in Pennsylvania.

Note: The Pennsylvania breast cancer dataset holds observations of breast cancer for 67 counties in the state of Pennsylvania. We were provided with geographic centroids, population, and cases. Without having demographics, then, we computed expected counts from a single rate applied to the entire region.

Question 1

For the Moran and Geary statistics experiment with the definition of weights and clearly report your findings.

- (a) *Examine the level of clustering in these data using Moran's statistic.*

To assess the level of overall clustering, we can evaluate Moran's test for spatial dependence. Recall that Moran's statistic calculates all pairwise products of the variable of interest, adjusting for spatial weights. We consider different weighting schemes and perform the test on the Pearson residuals obtained from both the unadjusted and the model that adjusts for both the Eastings and Northings. The results are summarized in Table 1. The results for the B, C, U, and minmax options for weightings were equivalent, so only those for B is shown. The results between the different weighting schemes do not vary greatly.

Weights	Unadjusted Residuals		Adjusted Residuals	
	I	p-value	I	p-value
Row Standardized (W)	0.223	0.001	0.167	0.007
Binary (B)	0.219	0.001	0.151	0.010
Variance Stabilizing (S)	0.221	0.001	0.159	0.008

Table 1: Moran's I statistic and p-value using Pearson's residuals from the unadjusted and spatially-adjusted models.

- (b) *Examine the level of clustering in these data using Geary's statistic.*

We use the de-trended residuals (from the adjusted model in the previous part) two compute the Geary's statistic using the various weights. The results are summarized in Table 2. Again, the results between the different weighting schemes do not vary greatly.

Weights	c	p-value
Row Standardized (W)	0.809	0.010
Binary (B)	0.807	0.025
Variance Stabilizing (S)	0.809	0.015

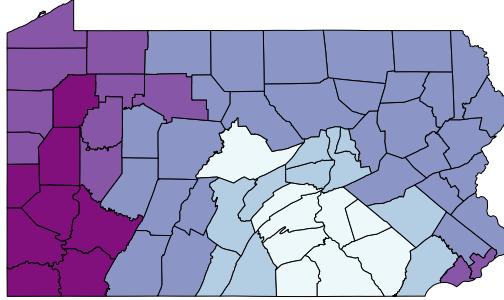
Table 2: Geary's statistic and p-value using Pearson's residuals from the spatially-adjusted model.

(c) Fit a Poisson lognormal-spatial smoothing model, with $Y_i|\mu_i \sim \text{Poisson}(N_i\mu_i)$, of the form

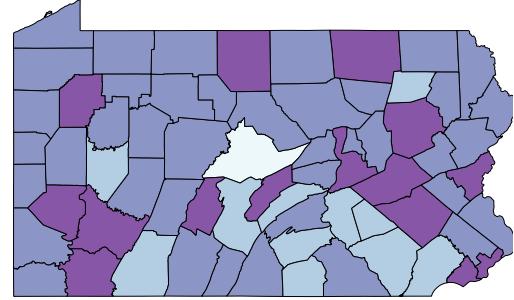
$$\log \mu_i = \alpha + V_i + U_i \quad (1)$$

with V_i and U_i non-spatial and ICAR spatial random effects, using `inla` with the default prior settings. Examine the level of clustering by looking at the random effects V_i and in particular their size in comparison with the U_i .

By comparing the empirical variance of U_i and the estimated variance of V_i , we find that approximately 47% of the variability in the data can be explained by the spatial variability. In Figure 1a, we see that there are two distinct regions with large estimated random effects, in the southwest and southeast portions of the state.



(a) Spatial Random Effects.



(b) Non-spatial Random Effects.

Figure 1: Estimated random effects for Pennsylvania breast cancer data.

Question 2

For each of the cluster detection methods, experiment with the different tuning parameters (circle sizes, numbers of cases, maximum size of population) and clearly report your findings.

- (a) Fit a Poisson log-linear model to these data and access the level of overdispersion in the data.

The overdispersion parameter is estimated as 3.71. To assess the significance of the overdispersion parameter κ , we performed a Monte Carlo test by simulating data from the null of Poisson distribution without overdispersion. Figure 2 is the histogram of κ estimates generated under the null, along with the estimated κ from the observed data. It clearly shows the excess Poisson variability in the observed data.

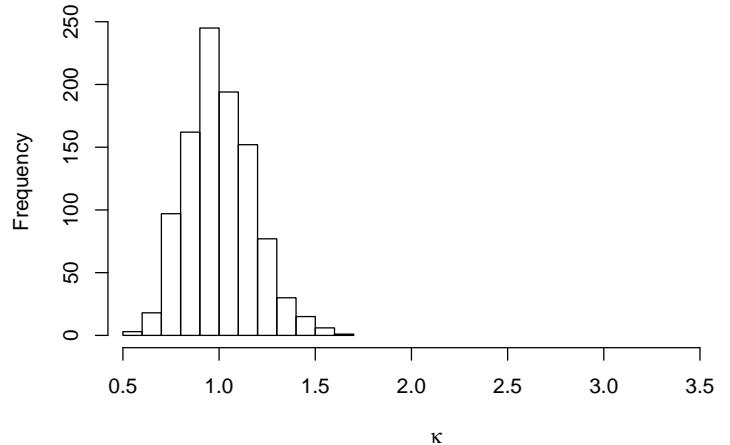


Figure 2: Histogram of κ estimates generated under the null, along with the estimate.

(b) Perform cluster detection on these data using the method of Openshaw.

Recall that the method of Openshaw slices the study region into a grid, and centers circles of a given size at each intersection. The observed cases relative to expected cases is assessed through a Poisson distribution, and nodes whose circles show p-values less than some threshold are flagged. This method involves a lot of dependent, frequentist hypothesis tests, which is always a frightening proposition.

Nevertheless, we performed cluster detection by the method of Openshaw with circle radius 30, 40 and 50, shown in Figure 3. The significance level for calling a cluster is 0.002. As the radius gets greater, we detected fewer clusters. All these figures suggest a cluster in the left bottom corner.

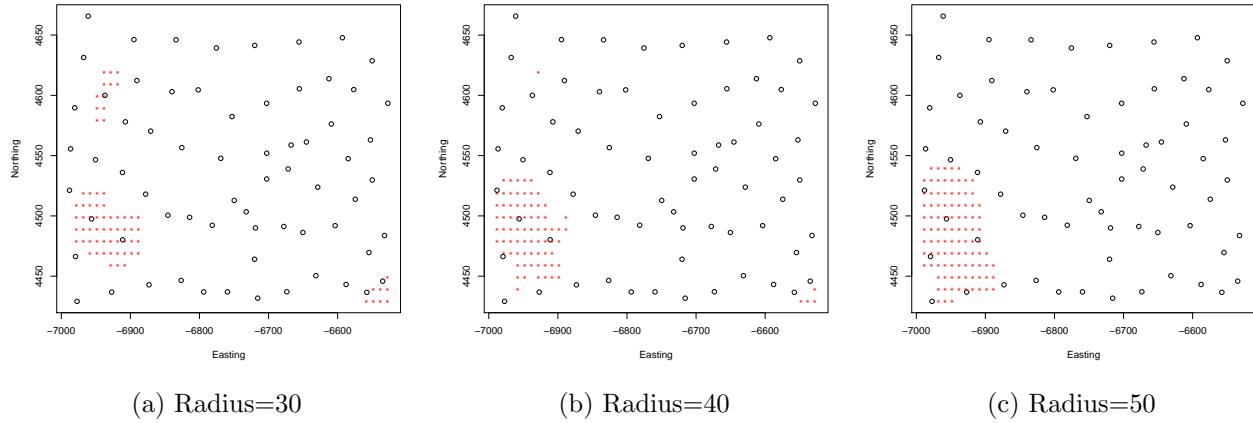


Figure 3: Cluster detection by the Openshaw method.

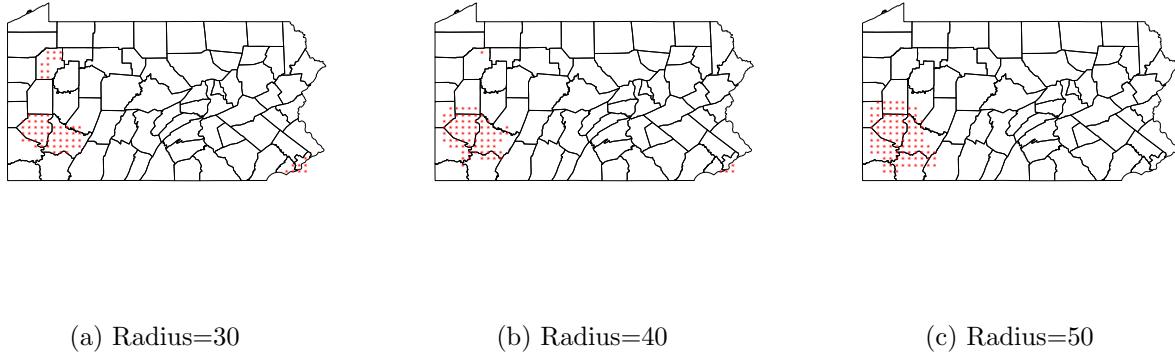
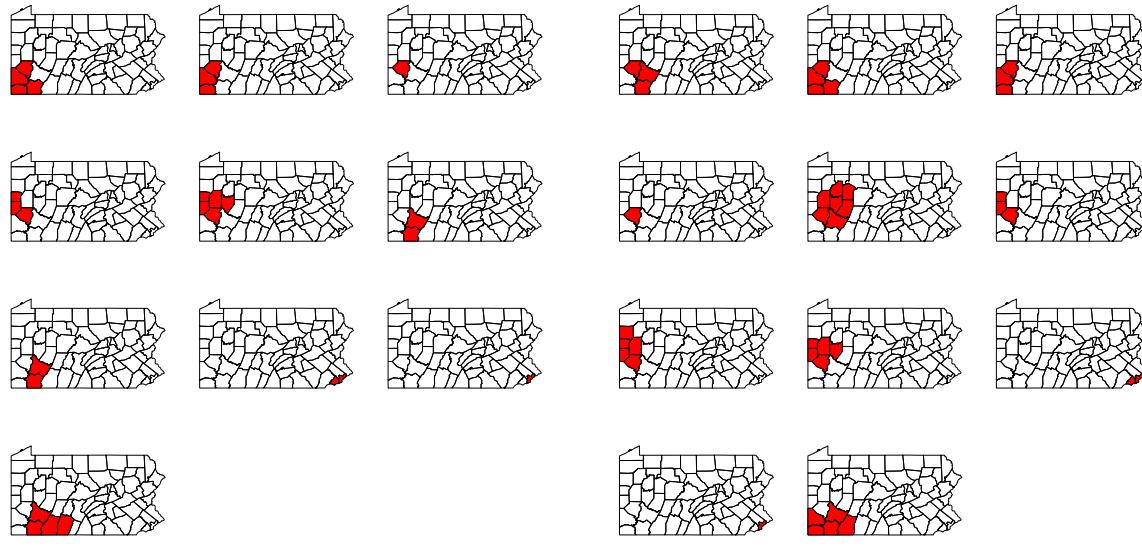


Figure 4: Cluster detection by the Openshaw method on the Pennsylvania map.

(c) Perform cluster detection on these data using the method of Besag and Newell.

The Besag Newell method considers circles containing k centroids, rather than the Openshaw method which considers circles of constant size. Figure 5 gives the results of cluster detection by the method of Besag and Newell, with number of cases 500 and 800, and significance level 0.05. As the number of cases increases, more clusters are detected.



(a) Number of cases = 500

(b) Number of cases = 800

Figure 5: Cluster detection by Besag and Newell's method.

(d) Perform cluster detection on these data using the method of Kulldorff and Nagarwalla.

Figure 6 shows the detected clusters using the method of Kulldorff and Nagarwalla with various maximum sizes of the population circles. The detected clusters are the same for circles containing a maximum of 50% and 20% of the population (see Figures 6a and 6b). As the maximum size of the population decreases, fewer clusters are detected. For the analyses where the circles contained a maximum of 15% and 10% of the population, we only detect one cluster (see Figures 6c and 6d). There are two counties that are found in all of the most likely clusters, regardless of maximum population size, which provides further evidence of a cluster in the southwest portion of Pennsylvania.

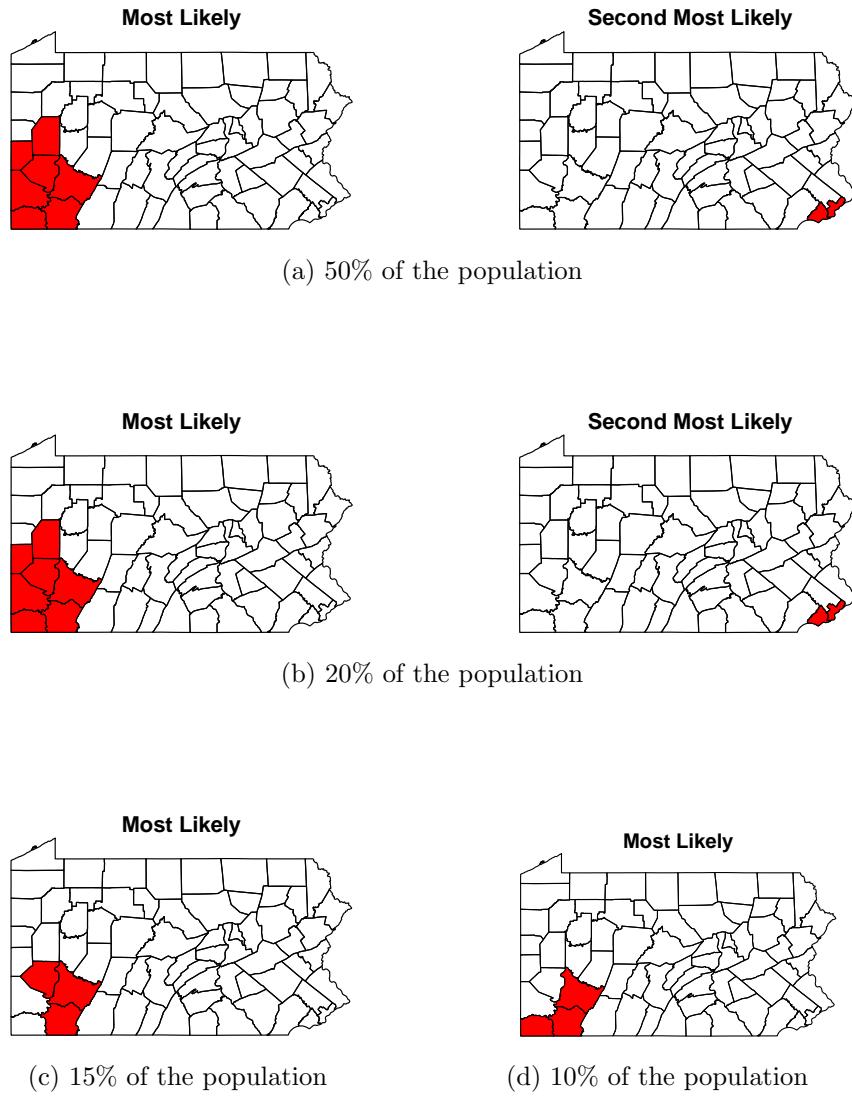


Figure 6: Cluster detection by Kulldorff and Nagarwalla's method.

- (e) Fit the model (1) and use this as a tool for cluster detection. Are there are disadvantages of using the model for this purpose?

The levels of spatial and non-spatial variability are the same order of magnitude, but the overall sizes of the random effects are not large. Generally speaking this approach does agree with the other cluster detection methods, suggesting that there may be a high area in the southwest corner. There are some clear disadvantages to using this smoothed approach, however. All extreme observations are smoothed towards the overall level. Thus, it may be very difficult to detect clusters, particularly in small sample settings.

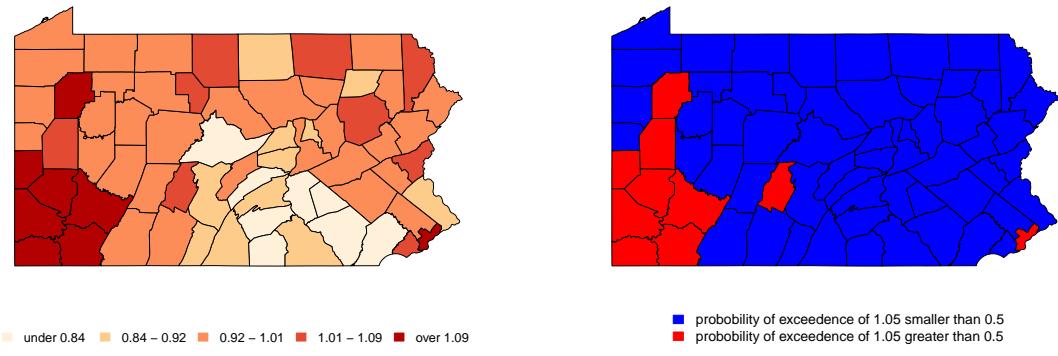


Figure 7: Cluster detection by Poisson lognormal smoothing model.

Appendix: R Code

```
library(xtable); library(spdep); library(INLA); library(sp); library(RColorBrewer); library(SpatialEpi); library(DCluster)

## Load Penn Data
penn.dat <- read.table("penn-data.txt", header=T, sep=",")

penn.dat$Expected <- penn.dat$pop*(sum(penn.dat$cases)/sum(penn.dat$pop))
penn.dat$Observed <- penn.dat$cases
penn.dat$Population <- penn.dat$pop

## Get Pennsylvania map file
load("USA_adm2.RData")
penn <- gadm[which(gadm$NAME_1=="Pennsylvania"), ]

## Get graph file in working format
penn.graph <- readLines("penn.graph")
n <- as.integer(penn.graph[1])
nb <- vector(mode="list", length=n)
for (i in 1:n) nb[[i]] <- as.integer(strsplit(penn.graph[i+1], " ")[[1]])[-(1:2)]
class(nb) <- "nb"
attr(nb, "region.id") <- 1:n
nb
#
# # Saved the nb object so I don't have to do this again!
# write.nb.gal(nb, "penn_nb")

# Read in nb file.
nb <- read.gal("penn_nb")
nb
```

Question 1

```
## Different weights
col.W <- nb2listw(nb, style="W", zero.policy=T)
col.B <- nb2listw(nb, style="B", zero.policy=T)
col.S <- nb2listw(nb, style="S", zero.policy=T)

quasipmod <- glm(Observed~1, offset=log(Expected), data=penn.dat, family="quasipoisson")
quasipmod2 <- glm(Observed~x+y, offset=log(Expected), data=penn.dat, family="quasipoisson")
resids <- residuals(quasipmod, type="pearson")
resids2 <- residuals(quasipmod2, type="pearson")

## Q 1 a: Moran I statistic
morW1 <- moran.test(resids, col.W)
morW2 <- moran.test(resids2, col.W)

morB1 <- moran.test(resids, col.B)
morB2 <- moran.test(resids2, col.B)

morS1 <- moran.test(resids, col.S)
morS2 <- moran.test(resids2, col.S)
```

```

#Make a table
tab1a <- data.frame(Weights = c("W", "B", "S"),
                      statUn = c(morW1$estimate[1], morB1$estimate[1], morS1$estimate[1]),
                      pval1 = c(morW1$p.value, morB1$p.value, morS1$p.value),
                      statAdj = c(morW2$estimate[1], morB2$estimate[1], morS2$estimate[1]),
                      pval2 = c(morW2$p.value, morB2$p.value, morS2$p.value)
)
print(xtable(tab1a, digits=3), include.rownames=F)

# Q 1 b: Geary's c Statistic #
gW <- geary.test(resids2, col.W)
gB <- geary.test(resids2, col.B)
gS <- geary.test(resids2, col.S)

tab1b <- data.frame(Weights = c("W", "B", "S"),
                      statUn = c(gW$estimate[1], gB$estimate[1], gS$estimate[1]),
                      pval1 = c(gW$p.value, gB$p.value, gS$p.value) )

print(xtable(tab1b, digits=3), include.rownames=F)

# Q 1 c: INLA #
penn.dat$ID <- 1:dim(penn.dat)[1]
penn.dat$ID2 <- penn.dat$ID
head(penn.dat)
inlaMod <- inla(Observed ~ 1 + f(ID, model="iid") + f(ID2, model="besag", graph="penn.graph"),
                  family = "poisson",
                  E=Expected,
                  data=penn.dat, control.predictor=list(compute=T))
summary(inlaMod)

REspatial <- inlaMod$summary.random$ID2[, 5]
var(REspatial)
range(REspatial)
# Variance of non-spatial random effects #
inlaMod$summary.hyperpar
REnonspatial <- inlaMod$summary.random$ID[, 5]
range(REnonspatial)

var(REspatial)/(var(REspatial) + 1/inlaMod$summary.hyperpar[1, 4])



## Make pretty pictures
nclr = 5 # 5 colors
plotclr <- brewer.pal(nclr,"BuPu") # Get some colors
brks <- c(seq(min(range(REspatial)), range(REnonspatial)),max(range(REspatial), range(REnonspatial)),
length.out=nclr+1)) # Want plots on similar scales


# plot Spatial Random Effects
plotval = REspatial # plotting variable
colornum <- findInterval(plotval,brks,all.inside=T)
colcode <- plotclr[colornum]

```

```

plot(penn, col=colcode)

# plot non-Spatial Random Effects
plotval = RENonspatial # plotting variable
colornum <- findInterval(plotval,brks,all.inside=T)
colcode <- plotclr[colornum]

plot(penn, col=colcode)
# leg = c(leglabs(round(brks,2)))
# legend("bottom", legend=leg,
#        fill=c(plotclr, NA),cex=0.75, horiz=T, border=plotclr, bty="n")

## Make legend
leg = c(leglabs(round(brks,2)))

plot.new()
legend("center", legend=leg,
       fill=c(plotclr, NA), cex=0.95, border=plotclr)

```

Question 2

```

### Write a helper function to estimate kappa
kappaaval <- function(Y, fitted, df){
  sum( (Y-fitted)^2/fitted )/df
}

# Q 2 a: Fit a Poisson log-linear model to these data and access the level of overdispersion in the data
mod <- glm(Observed ~ 1, offset=log(Expected), data=penn.dat, family="quasipoisson")
summary(mod) # overdispersion parameter is estiamted to be 3.71
# Note, can estimate kappa using summary(mod)$dispersion
kappaEst <- kappaaval(penn.dat$Observed, mod$fitted, mod$df.resid)

## Assessing significance of overdisperion using Monte Carlo permutation
nMC <- 1000
nAreas <- dim(penn.dat)[1]
yMC <- matrix(rpois(n = nMC*nAreas, lambda=penn.dat$Expected), nrow=nAreas, ncol=nMC)
kappaMC <- NULL
for (i in 1:nMC){
  modMC <- glm(yMC[,i] ~ 1, offset = log(penn.dat$Expected), family="quasipoisson")
  kappaMC[i] <- summary(modMC)$dispersion
}

pdf("Homework/HW05/2a_overdispersion.pdf", height=4.5, width=6.5)
hist(kappaMC, xlim=c(min(kappaMC), max(kappaMC, kappaEst)), main="", xlab=expression(kappa))
abline(v = kappaEst, col="red")
dev.off()

# Q 2 b: Cluster detection by the method of Openshaw (circle size) #
# cluster detection using the method of Openshaw

```

```

penn.gam30 <- opgam(data=penn.dat, radius=30, step=10, alpha=.002)

plot(penn.dat$x, penn.dat$y, xlab="Easting", ylab="Northing")
points(penn.gam30$x, penn.gam30$y, col="red", pch="*")

penn.gam40 <- opgam(data=penn.dat, radius=40, step=10, alpha=.002)

plot(penn.dat$x, penn.dat$y,xlab="Easting", ylab="Northing")
points(penn.gam40$x, penn.gam40$y, col="red", pch="*")

penn.gam50 <- opgam(data=penn.dat, radius=50, step=10, alpha=.002)

plot(penn.dat$x, penn.dat$y,xlab="Easting", ylab="Northing")
points(penn.gam50$x, penn.gam50$y, col="red", pch="*")

# Q 2 c: Cluster detection by the method of Besag and Newell (k cases) #
# cluster detection using the method of Besag and Newell
## k = 500
library(SpatialEpi)
penn.bn500 <- besag_newell(penn.dat[, c("x", "y")], population=penn.dat$Population,
    cases=penn.dat$Observed, k=500, alpha.level=0.01)

ids <- rep(NA, length(penn.bn500$clusters))

BNSig <- length(penn.bn500$p.values[penn.bn500$p.values<0.01])
resmat <- matrix(NA, nrow=BNSig, ncol=67); reslen <- NULL

for(i in 1:length(penn.bn500$clusters)){
  reslen[i] <- length(penn.bn500$clusters[[i]]$location.IDs.included)
  resmat[i, 1:reslen[i]] <- penn.bn500$clusters[[i]]$location.IDs.included
}

par(mfrow=c(4, 3), mar=c(0, 1, 0, 1), oma=c(0, 1, 0, 1))
for(i in 1:10){
  plot(penn)
  plot(penn[resmat[i, ]![is.na(resmat[i, ])]], ], col="red", add=T)
}

## k = 800
penn.bn800 <- besag_newell(penn.dat[, c("x", "y")], population=penn.dat$Population,
    cases=penn.dat$Observed, k=800, alpha.level=0.01)

BNSig <- length(penn.bn800$p.values[penn.bn800$p.values<0.01])
BNSig
resmat <- matrix(NA, nrow=BNSig, ncol=67); reslen <- NULL

for(i in 1:length(penn.bn800$clusters)){
  reslen[i] <- length(penn.bn800$clusters[[i]]$location.IDs.included)
  resmat[i, 1:reslen[i]] <- penn.bn800$clusters[[i]]$location.IDs.included
}

```

```

par(mfrow=c(4, 3), mar=c(0, 1, 0, 1), oma=c(0, 2, 0, 0))
for(i in 1:11){
  plot(penn)
  plot(penn[resmat[i, ]![is.na(resmat[i, ])]], col="red", add=T)
}

# Q 2 d: Cluster detection by the method of Kulldorff and Nagarwalla (Fraction of population) #
Kpoisson08 <- kulldorff(penn.dat[, c("x", "y")], population=penn.dat$Population,
  cases=penn.dat$Observed, pop.upper bound=0.8, n.simulations=9999, alpha.level=0.05, plot=T)

par(mfrow=c(1,1))
Kclust08 <- Kpoisson08$most.likely.cluster$location.IDs.included
K2clust08 <- Kpoisson08$secondary.clusters[[1]]$location.IDs.included
# K3clust08 <- Kpoisson08$secondary.clusters[[2]]$location.IDs.included

par(mfrow=c(1, 2))
plot(penn)
title("Most Likely", line=-2)
plot(penn[Kclust08, ], col="red", add=T)
plot(penn)
title("Second Most Likely", line=-2)
plot(penn[K2clust08, ], col="red", add=T)

Kpoisson02 <- kulldorff(penn.dat[, c("x", "y")], population=penn.dat$Population,
  cases=penn.dat$Observed, pop.upper bound=0.2, n.simulations=9999, alpha.level=0.05, plot=T)
Kclust02 <- Kpoisson02$most.likely.cluster$location.IDs.included
K2clust02 <- Kpoisson02$secondary.clusters[[1]]$location.IDs.included

par(mfrow=c(1, 2))
plot(penn)
title("Most Likely", line=-2)
plot(penn[Kclust02, ], col="red", add=T)
plot(penn)
title("Second Most Likely", line=-2)
plot(penn[K2clust02, ], col="red", add=T)

Kpoisson05 <- kulldorff(penn.dat[, c("x", "y")], population=penn.dat$Population,
  cases=penn.dat$Observed, pop.upper bound=0.5, n.simulations=9999, alpha.level=0.05, plot=T)
Kclust05 <- Kpoisson05$most.likely.cluster$location.IDs.included
K2clust05 <- Kpoisson05$secondary.clusters[[1]]$location.IDs.included

par(mfrow=c(1, 2))
plot(penn)
title("Most Likely", line=-2)
plot(penn[Kclust05, ], col="red", add=T)
plot(penn)
title("Second Most Likely", line=-2)
plot(penn[K2clust05, ], col="red", add=T)

Kpoisson01 <- kulldorff(penn.dat[, c("x", "y")], population=penn.dat$Population,
  cases=penn.dat$Observed, pop.upper bound=0.1, n.simulations=9999, alpha.level=0.05, plot=T)

```

```

Kclust01 <- Kpoisson01$most.likely.cluster$location.IDs.included

Kpoisson015 <- kulldorff(penn.dat[, c("x", "y")], population=penn.dat$Population,
  cases=penn.dat$Observed, pop.upper bound=0.15, n.simulations=9999, alpha.level=0.05, plot=T)
Kclust015 <- Kpoisson015$most.likely.cluster$location.IDs.included

plot(penn)
title("Most Likely", line=-2)
plot(penn[Kclust01, ], col="#FF0000", add=T)

plot(penn)
title("Most Likely", line=-2)
plot(penn[Kclust015, ], col="red", add=T)

# Q 2 e: Poisson lognormal smoothing model #
RR <- inlaMod$summary.fitted.values[,1]
nclr <- 5
plotval <- RR
plotclr <- brewer.pal(nclr,"OrRd")
brks <- c(seq(min(plotval),max(plotval), length.out=nclr+1))
colornum <- findInterval(plotval,brks,all.inside=T)
colcode <- plotclr[colornum]

# plot points

# plot(penn.dat$x, penn.dat$y, axe=T, pch=19, col=colcode, cex=2, xlab="Easting", ylab="Northing")
plot(penn, col=colcode)
leg = c(leglabs(round(brks,2)))
legend("bottom", legend=leg,
  fill=c(plotclr, NA),cex=0.95, horiz=T, border=plotclr, bty="n")

# plot areas with medians above 1.05 #
RR05 = inlaMod$summary.fitted.values[,4]>1.05
nclr = 2
plotval = RR05
plotclr <- c("blue","red")
brks <- c(seq(min(plotval),max(plotval), length.out=nclr+1))
colornum <- findInterval(plotval,brks,all.inside=T)
colcode <- plotclr[colornum]

# plot(penn.dat$x, penn.dat$y, axe=T, pch=19, col=colcode, cex=2, xlab="Easting", ylab="Northing")
plot(penn, col=colcode)
leg = c("probability of exceedence of 1.05 smaller than 0.5","probability of exceedence of 1.05 greater than 0.5" )
legend("bottom", legend=leg, fill=c(plotclr, NA),cex=1,ncol=1, border=plotclr, bty="n")

```