Bios 737 Project

Hannah Waddel

Did some work with Rachel Parker, Victoria Kennerly, Sarita Mohanty, and Nikita Thomas.

Introduction

In the Black Mesa, Arizona area, the Peabody Coal Mining company has obtained a lease to mine for coal. Following a required archaeological survey of the Ancestral Pueblan (Anasazi) settlements in the area, we have obtained location information on around 700 settlements. We are examining 490 settlements, dated between 850 and 1050 C.E. The settlements can be divided into older settlements and newer settlements. The older settlements are dated between 850 and 949 C.E. while the newer settlements are dated between 950 and 1050 C.E.

The Ancestral Pueblans experienced rapid population growth around 1000 C.E. which was followed by a precipitous drop in population and subsequent abandonment of many settlements around 1050 C.E. This informs our interest in settlements between 850 and 1050, as well as our division of settlements into old and new. We are interested in analyzing and comparing the spatial distribution of the old and new settlements.

Methods

In order to compare the spatial distribution of the two sets of settlements, we will first assume that old and new settlements are spatial Poisson processes with intensities $\lambda(s)$, where s is a location within the study area. The number of events in an area A is distributed as $N(A) \sim \text{Poisson}(\lambda(s)\ |A|)$, where $|A|$ is the area of A. To estimate the intensities of each process, we will fit the Kernel Density Estimate (KDE) of each set of settlements.

We will then estimate the log relative risk surface of the old and new settlements. This will allow us to analyze where their intensities, and thus the spatial patterns, may differ. Extreme values of r(s) will be evaluated at the $\alpha=0.05$ level with inference based on Monte Carlo testing with n=999 simulations under the random labelling null hypothesis. The integrated squared deviation from null (0) value will be calculated for the observed data as well as the simulated data, and inference again based on Monte Carlo testing.

In order to determine whether the sites appear to be uniformly randomly distributed in the study area, as well as the spatial scale of any clustering or regularity, we will fit Ripley's K/L function to the old, new, and all settlements in order to detect the scale at which we observe clustering. We will estimate p-values using Monte Carlo testing with n=999 simulations under the complete spatial randomness null hypothesis.

Results

Fig. 1. Map of all settlements

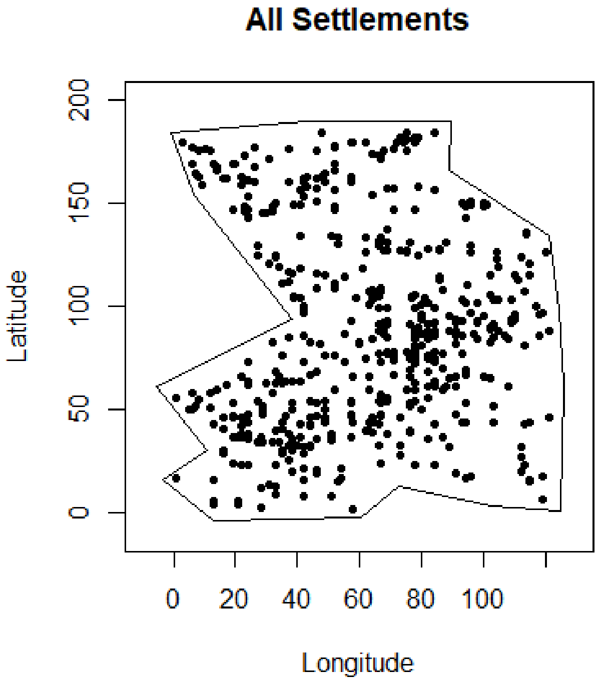


Fig. 2. Map and kernel density estimate of old settlements

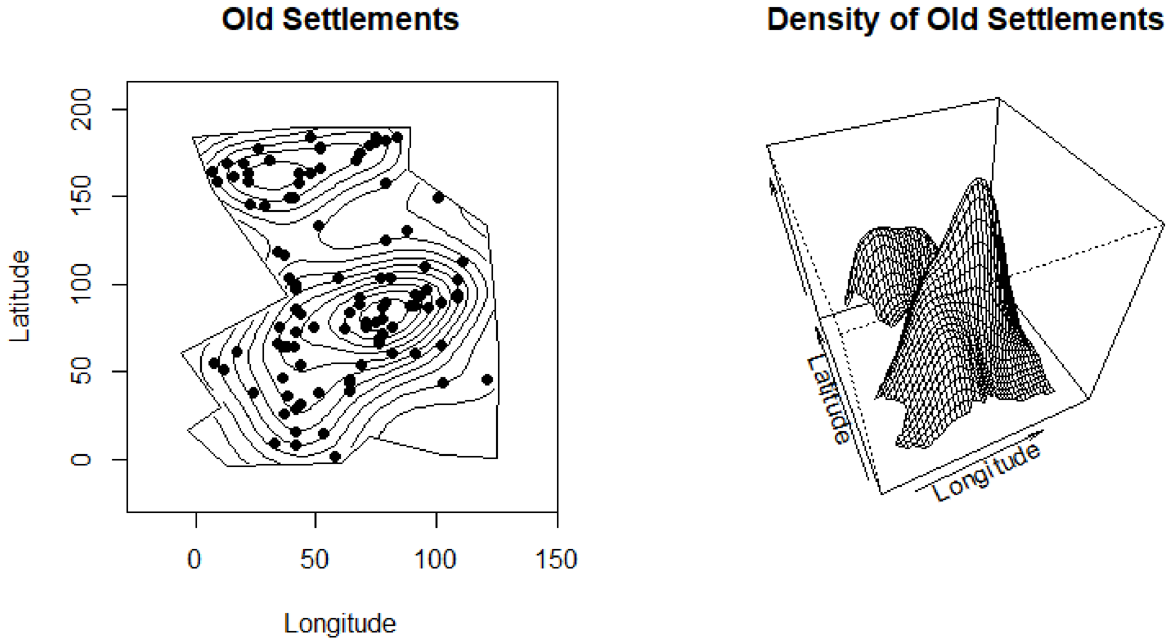


Fig. 3. Map and kernel density estimate of new settlements

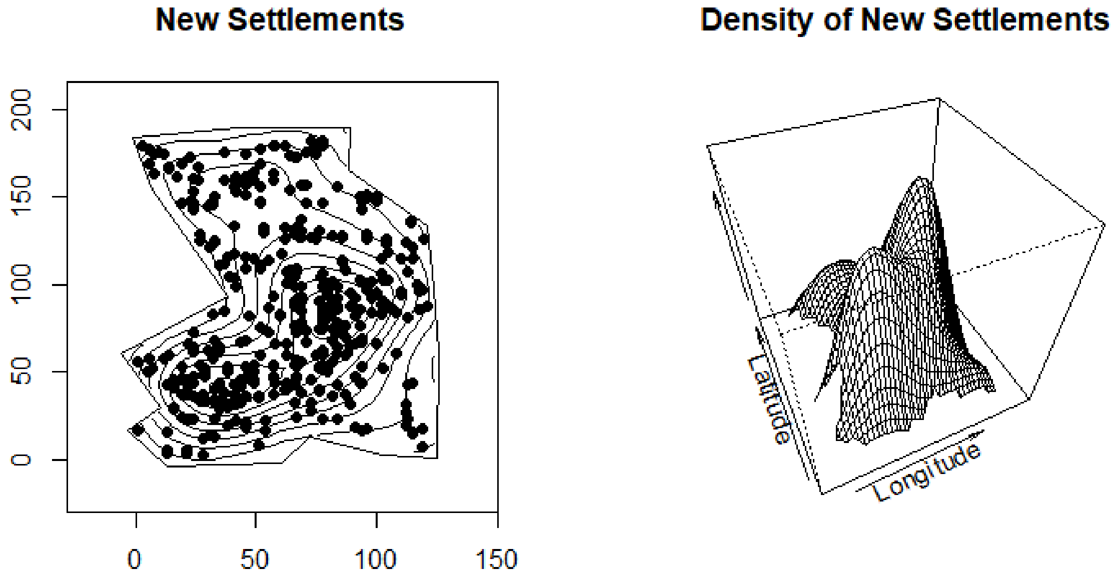


Fig. 4. Contour plot of log relative risk surface of old versus new settlements

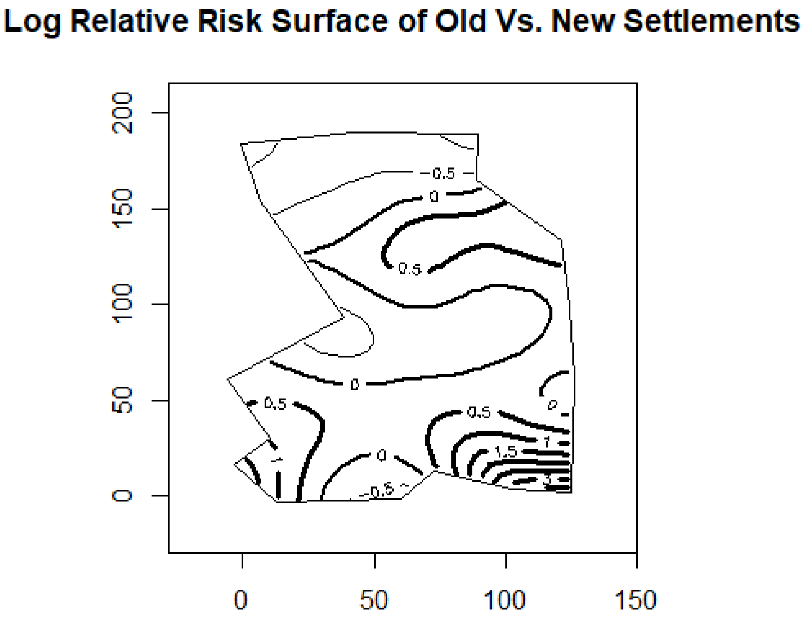


Fig. 5. Histogram of r(s) test statistic, integrated squared distance from 0

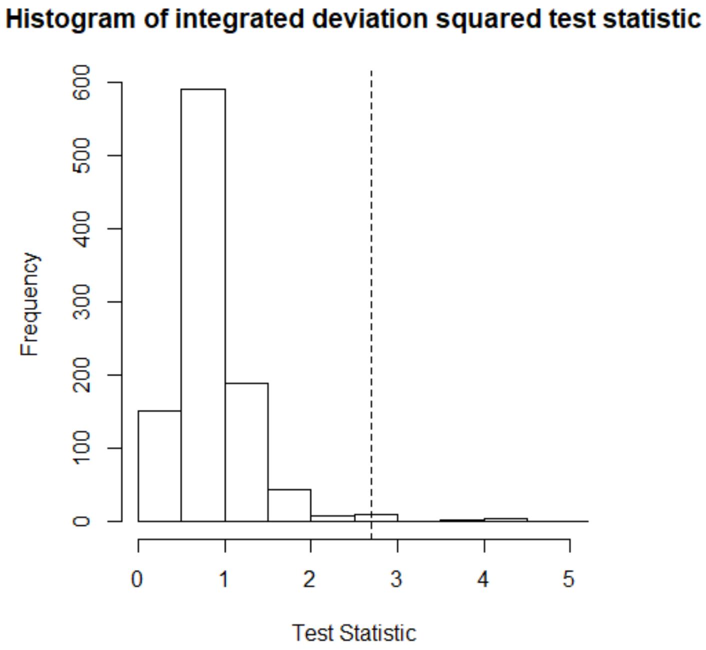


Fig. 6. Log relative risk surface of old versus new settlements, with - signs indicating a value below the Monte Carlo 2.5th percentile and + signs indicating a value above the Monte Carlo 97.5th percentile.

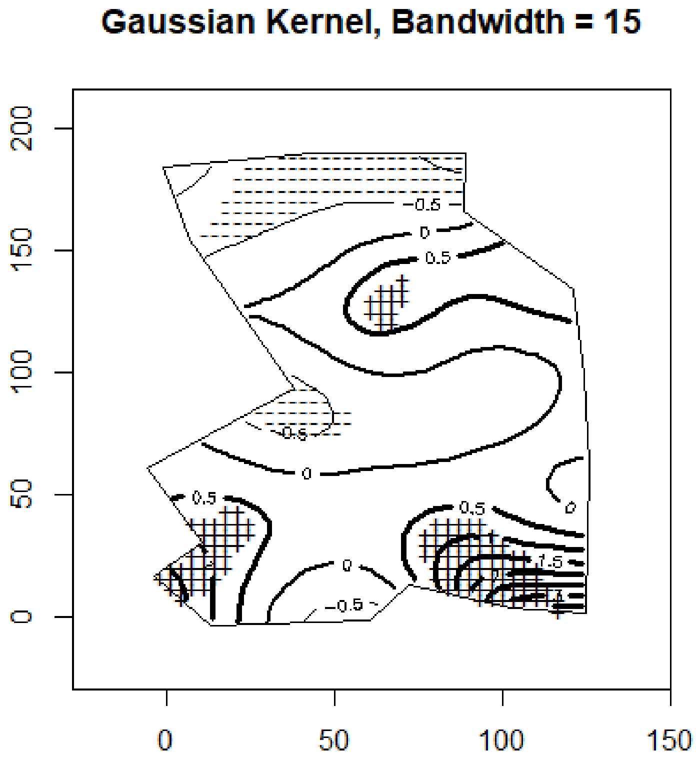
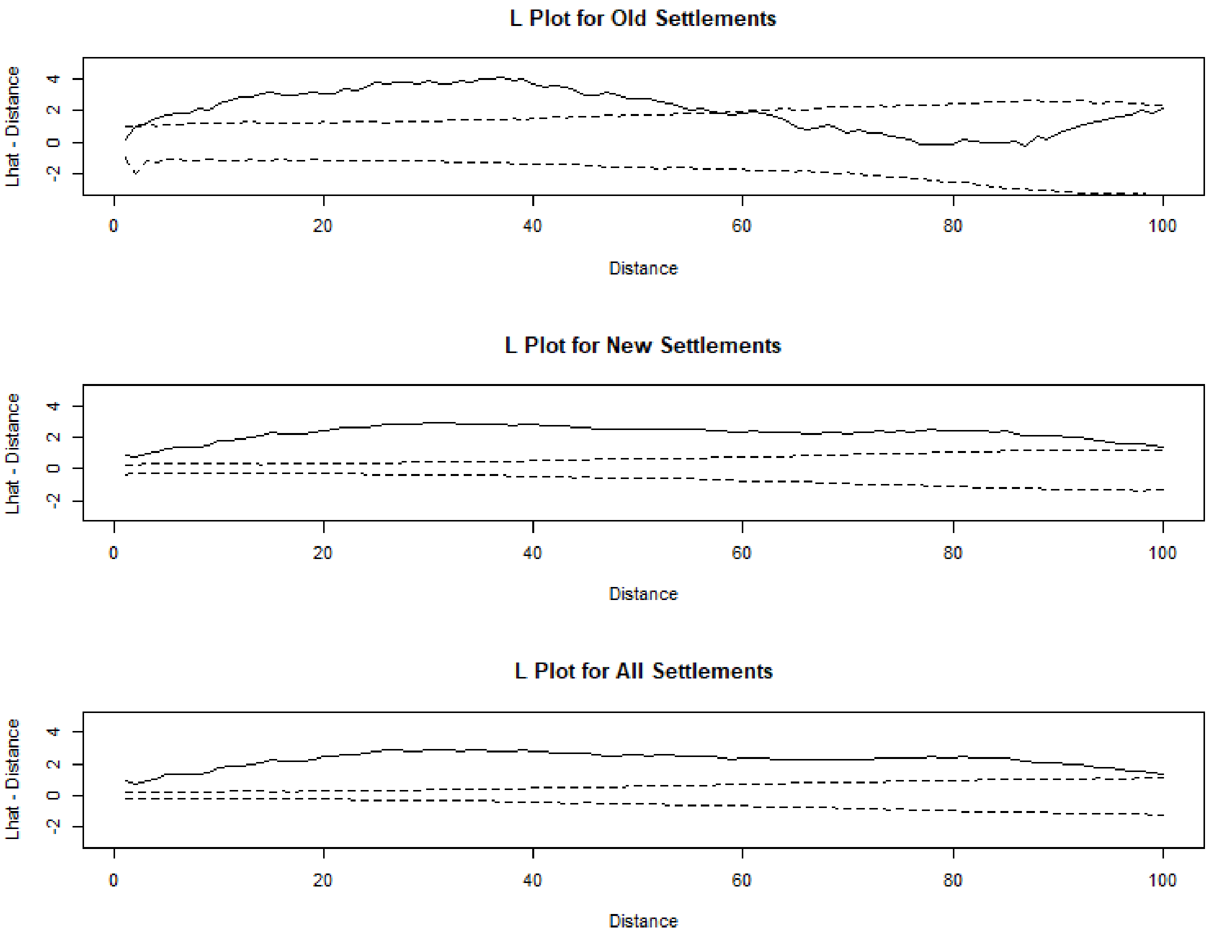


Fig. 7. L Plots for old, new, and all settlements. Dashed lines indicate 2.5th and 97.5th percentiles from Monte Carlo simulations.



L Plot for Old Settlements

L Plot for New Settlements

L Plot for All Settlements

Conclusions

The kernel density estimates seem to indicate peaks toward the center of the study area, indicating a higher concentration of settlements there. The log relative risk surface peaks in the southeast corner of the study area, with minima near the west and north areas of the study area. This indicates potential differences in the density of old and new settlements in those areas.

The log relative risk surface is below the 2.5th percentile in some north and central areas. This would indicate a lower than expected ratio of new to old settlements, where there are fewer new settlements for each old settlement than we would expect. The log relative risk surface seems to be above the 97.5th percentile in some of the southern areas, indicating places with a higher than expected ratio of new to old settlements. Similarly, the integrated squared deviation from null of the observed data is observed to be above the 97.5th percentile of the simulated data. This indicates clustering, as it seems old and new settlements cluster together within each group.

The L functions for old settlements, new settlements, and all settlements indicate that there is significant evidence that the data do not follow a complete spatial random pattern. For old settlements, we observe clustering at distances less than 60 degrees. For new settlements, we observe clustering at distances less than 100 degrees. We observe the same pattern of clustering for all settlements as well. Since the old settlements and new settlements tended to cluster, this may be due to new areas that became habitable with a change in climate or some other environmental or cultural change.

Appendix

```r
##Bios 737 Project
#Set up libraries
library(splancs)
library(KernSmooth)

#Read in the data
setwd("~/Classes/BIOS 737/Project")
old.anasazi <- read.csv("anasazi.old.csv",header=T,stringsAsFactors = F)
new.anasazi <- read.csv("anasazi.new.csv",header=T,stringsAsFactors = F)
all.anasazi <- rbind(old.anasazi,new.anasazi)

par(pty="s",mfrow=c(1,1))
plot(all.anasazi,xlim=c(-10,130),ylim=c(-10,200),pch=20,main="All Settlements",xlab
="Longitude",ylab="Latitude")

# mypoly <- getpoly()
#
# write.csv(mypoly,file="projpoly.csv")

mypoly <- read.csv("projpoly.csv",header=T,stringsAsFactors = F)
names(mypoly) <- c("x","y")
polygon(mypoly)

old.bw <- 15
new.bw <- 15

expand.amt <- 1.5

xmin <- min(old.anasazi$long,new.anasazi$long)-expand.amt*old.bw
xmax <- max(old.anasazi$long,new.anasazi$long)+expand.amt*old.bw
ymin <- min(old.anasazi$lat,new.anasazi$lat)-expand.amt*old.bw
ymax <- max(old.anasazi$lat,new.anasazi$lat)+expand.amt*old.bw

old.bd <- list(c(xmin,xmax),c(ymin,ymax))
new.bd <- list(c(xmin,xmax),c(ymin,ymax))

grid.n <- 51

par(pty="s",mfrow=c(1,2))
old.dens <- bkde2D(old.anasazi,old.bw,gridsize=c(grid.n,grid.n),old.bd)
new.dens <- bkde2D(new.anasazi,new.bw,gridsize = c(grid.n,grid.n),new.bd)

grid.x <- seq(from=xmin,to=xmax,length.out = grid.n)
grid.y <- seq(from=ymin,to=ymax,length.out = grid.n)

grid.all <- expand.grid(grid.x,grid.y)
names(grid.all) <- c("x","y")
```

```
inside <- inout(grid.all,mypoly)

old.dens$fhat[!inside] <- NA
new.dens$fhat[!inside] <- NA

#Plot the density and points
par(mfrow=c(1,2))
contour(old.dens$x1,old.dens$x2,old.dens$fhat,main="Old Settlements",drawlabels = F
,xlab="Longitude",ylab="Latitude")
polygon(mypoly)
points(old.anasazi,pch=19)

persp(old.dens$x1,old.dens$x2,old.dens$fhat,theta=-25,phi=45,xlab="Longitude",ylab=
"Latitude",main="Density of Old Settlements",zlab="")

contour(new.dens$x1,new.dens$x2,new.dens$fhat,main="New Settlements",drawlabels = F
)
polygon(mypoly)
points(new.anasazi,pch=19)

persp(new.dens$x1,new.dens$x2,new.dens$fhat,theta=-25,phi=45,xlab="Longitude",ylab=
"Latitude",main="Density of New Settlements",zlab="")


#Define relative risk surface
log.risk <- log(new.dens$fhat/old.dens$fhat)

log.risk[!is.finite(log.risk)] <- NA

par(mfrow=c(1,1))
contour(old.dens$x1,old.dens$x2,log.risk,
        levels=c(-1,-.5,0,.5,1,1.5,2,2.5,3,3.5),
        lwd=c(1,1,2,3,3,3,3,3,3,3),
        main="Log Relative Risk Surface of Old Vs. New Settlements"
        )
polygon(mypoly)

#persp(old.dens$x1,old.dens$x2,log.risk,theta=-25,phi=45,xlab="Longitude",ylab="Lat
itude",main="Log Relative Risk Surface",zlab="")

#Calculate test statistic for data
data.test.stat <- sum( (log.risk[is.finite(log.risk) & !is.na(log.risk)]/(diff(old.
dens$x1)[1]*diff(old.dens$x2)[1]))^2 )

#Simulations to do this

n.sim <- 999
n.old <- nrow(old.anasazi)
n.new <- nrow(new.anasazi)
n.all <- nrow(all.anasazi)

#Compare Risk Surface to Random Labeling risk surface
ind <- 1:n.all
```

```r
#Create list to store test statistics
sim.test.stat <- rep(NA,n.sim)

#Create list to store risk surfaces
sim.rr <- list()

for(i in 1:n.sim){
  #Generate Random labels
  old.lab <- sample(ind,size=n.old)
  old.x <- all.anasazi$long[old.lab]
  old.y <- all.anasazi$lat[old.lab]
  old.pts <- as.points(old.x,old.y)

  new.lab <- ind[-old.lab]
  new.x <- all.anasazi$long[new.lab]
  new.y <- all.anasazi$lat[new.lab]
  new.pts <- as.points(new.x,new.y)

  old.rand <- bkde2D(old.pts,old.bw,gridsize=c(grid.n,grid.n),old.bd)
  new.rand <- bkde2D(new.pts,new.bw,gridsize=c(grid.n,grid.n),new.bd)

  old.rand$fhat[!inside] <- NA
  new.rand$fhat[!inside] <- NA

  ratio.rand <- log(new.rand$fhat/old.rand$fhat)

  sim.rr[[i]] <- ratio.rand

  sim.test.stat[i] <- sum( (ratio.rand[is.finite(ratio.rand) & !is.na(ratio.rand)]/
(diff(old.rand$x1)[1]*diff(old.rand$x2)[1]))^2 )
}

hist(sort(sim.test.stat), main="Histogram of r test statistic",xlab="Test Statistic
")
abline(v=data.test.stat,lty=2)

#Find envelopes for the simulated Risk Ratios
#Find 97.5 quantile
q97.5 <- matrix(data=NA,nrow=grid.n,ncol=grid.n)
this.point.rr <- rep(NA,n.sim)
for(i in 1:grid.n){
  for(j in 1:grid.n){
    for(k in 1:n.sim){
      this.point.rr[k] <- sim.rr[[k]][i,j]
    }
  q97.5[i,j] <- quantile(this.point.rr,probs=.975,na.rm=T)
  }
}

#find 2.5 quantile
q2.5 <- matrix(data=NA,nrow=grid.n,ncol=grid.n)
this.point.rr <- rep(NA,n.sim)
```

```r
for(i in 1:grid.n){
  for(j in 1:grid.n){
    for(k in 1:n.sim){
      this.point.rr[k] <- sim.rr[[k]][i,j]
    }
    q2.5[i,j] <- quantile(this.point.rr,probs=.025,na.rm=T)
  }
}

#Plot where the data's log relative risk surface is outside the envelopes
contour(old.dens$x1,old.dens$x2,log.risk,
        levels=c(-1,-.5,0,.5,1,1.5,2,2.5,3,3.5),
        lwd=c(1,1,2,3,3,3,3,3,3,3),
        main="Gaussian Kernel, Bandwidth = 15"
)
polygon(mypoly)
above <- which(log.risk>q97.5,arr.ind=T)
points(old.dens$x1[above[,1]],old.dens$x2[above[,2]],pch="+")

below <- which(log.risk<q2.5,arr.ind=T)
points(old.dens$x1[below[,1]],old.dens$x2[below[,2]],pch="-")


#Find the K function of the data
#distances at which to calculate k
dists <- seq(from=1,to=100,by=1)

#Find the K function of the old settlements
names(old.anasazi) <- c("x","y")
p.old.anasazi <- as.points(old.anasazi)
khat.old <- khat(p.old.anasazi,as.points(mypoly),dists)

#Find the K function of the new settlements
names(new.anasazi) <- c("x","y")
p.new.anasazi <- as.points(new.anasazi)
khat.new <- khat(p.new.anasazi,as.points(mypoly),dists)

#Find the K function of all the settlements
names(all.anasazi) <- c("x","y")
p.all.anasazi <- as.points(all.anasazi)
khat.all <- khat(p.all.anasazi,as.points(mypoly),dists)

#Simulate CSR Data, find K functions
khat.sim.old <- matrix(data=NA,nrow=n.sim,ncol=length(dists))
khat.sim.new <- matrix(data=NA,nrow=n.sim,ncol=length(dists))
khat.sim.all <- matrix(data=NA,nrow=n.sim,ncol=length(dists))

for(i in 1:n.sim){
  old.csr <- csr(as.points(mypoly),n.old)
  khat.sim.old[i,] <- khat(old.csr,as.points(mypoly),dists)

  new.csr <- csr(as.points(mypoly),n.new)
  khat.sim.new[i,] <- khat(new.csr,as.points(mypoly),dists)
  all.csr <- csr(as.points(mypoly),n.all)
  khat.sim.all[i,] <- khat(all.csr,as.points(mypoly),dists)

  if(i %% 50 == 0){
    print(i)
  }
}

khat.old.q2.5 <- apply(khat.sim.old,2,quantile,probs=.025,names=F)
khat.old.q97.5 <- apply(khat.sim.old,2,quantile,probs=.975,names=F)

khat.new.q2.5 <- apply(khat.sim.new,2,quantile,probs=.025,names=F)
khat.new.q97.5 <- apply(khat.sim.new,2,quantile,probs=.975,names=F)

khat.all.q2.5 <- apply(khat.sim.all,2,quantile,probs=.025,names=F)
khat.all.q97.5 <- apply(khat.sim.all,2,quantile,probs=.975,names=F)


##Calculate L functions for everything
lhat.old <- sqrt(khat.old/pi)
lhat.new <- sqrt(khat.new/pi)
lhat.all <- sqrt(khat.all/pi)

lhat.old.q2.5 <- sqrt(khat.old.q2.5/pi)
lhat.old.q97.5 <- sqrt(khat.old.q97.5/pi)

lhat.new.q2.5 <- sqrt(khat.new.q2.5/pi)
lhat.new.q97.5 <- sqrt(khat.new.q97.5/pi)

lhat.all.q2.5 <- sqrt(khat.all.q2.5/pi)
lhat.all.q97.5 <- sqrt(khat.all.q97.5/pi)

#Plot L functions for everything
par(mfrow=c(3,1),pty="m")

plot(dists,lhat.old-dists,ylim=c(-3,5),main="L Plot for Old Settlements",xlab="Distance",ylab="Lhat - Distance",type='n')
lines(dists,lhat.old-dists,lty=1)
lines(dists,lhat.old.q2.5-dists,lty=2)
lines(dists,lhat.old.q97.5-dists,lty=2)

plot(dists,lhat.new-dists,ylim=c(-3,5),main="L Plot for New Settlements",xlab="Distance",ylab="Lhat - Distance",type='n')
lines(dists,lhat.new-dists,lty=1)
lines(dists,lhat.new.q2.5-dists,lty=2)
lines(dists,lhat.new.q97.5-dists,lty=2)

plot(dists,lhat.all-dists,ylim=c(-3,5),main="L Plot for All Settlements",xlab="Distance",ylab="Lhat - Distance",type='n')
lines(dists,lhat.new-dists,lty=1)
lines(dists,lhat.all.q2.5-dists,lty=2)
lines(dists,lhat.all.q97.5-dists,lty=2)
```