

##### point pattern analysis #####

#### in this example we are using three different packages: "spatstat", "maptools", "sp", "lattice", and "GISTools"

### the format that "spatstat" store spatial point data is as 'ppp' format, which is different than what we have seen in "GISTools"

### easily we can convert the 'ppp' format into 'SpatialPoints' or vice versa

#### install and load packages

## if you have installed packages before, you do not need to reinstall them

```
>install.packages(c("spatstat", "sp", "GISTools", "maptools", "lattice"))
```

```
>library(spatstat)
```

```
>library(sp)
```

```
>library(GISTools)
```

```
>library(maptools)
```

```
>library(lattice)
```

#### load the data

```
>data("japanesepines")
```

#### let's look at the data in more details

```
>summary(japanesepines)
```

#### the "japanesepines" is in ppp format. Let's convert it into "SpatialPoints"

```
>spjpines <- as(japanesepines, "SpatialPoints")
```

## now let's look at it in more details again

```
>summary(spjpines)
```

#### just for fun!!! let's turn it back into 'ppp' format

```
>jp.ppp <- as(spjpines, "ppp")
```

### see how the details are

```
summary(jp.ppp)
```

#### oh oh ... the window size and area is not as same as the initial data set

## Let's make it correct

```
>jp.ppp$window <- japanesepines$window
```

### now let's see if it is fixed

```
>summary(jp.ppp)
```

#### load more data sets and convert them into 'SpatialPoints'

```
>data("redwoodfull")
```

```
>spred <- as(redwoodfull, "SpatialPoints")
```

```
>data("cells")
```

```
>spcells <- as(cells, "SpatialPoints")
```

### let's explore the new data sets

```
>summary(spred)
```

```
>summary(spcells)
```

### for the 'japanesepines' when it convert to 'SpatialPoints', the 'max' values are 5.7. While for 'redwoodfull' and 'cells' the max values are 1.

## to scale all data sets same and make the fit, we will use 'elide' demand

```
>spjpines1 <- elide(spjpines, scale=T, unitsq=T)
```

#### now let's see if it fixed

```
>summary(spjpines1)
```

#### combine all three data sets into a single data frame

```
>dpp<-data.frame(rbind(coordinates(spjpines1), coordinates(spred),  
                      coordinates(spcells)))
```

#### let's make the length of each data set as a code that we can call them later on on the data frame

```
>njap<-nrow(coordinates(spjpines1))
```

```
>nred<-nrow(coordinates(spred))
```

```
>ncells<-nrow(coordinates(spcells))
```

### rebuilt the data frame from the previous one and add another column next to it to say which point has come from which data set

```
>dpp<-cbind(dpp,c(rep("JAPANESE",njap), rep("REDWOOD", nred), rep("CELLS", ncells)))
```

### name the columns on data frame

```
>names(dpp)<-c("x", "y", "DATASET")
```

### let's look into the data frame

```
>summary(dpp)
```

##### here we are going to use 'xyplot' from the 'lattice' package to bivariate scatterplots for each value of datasets

```
>xyplot(y~x|DATASET, data=dpp, pch=19, aspect=1)
```

##### G function #####

### run the G test on the pines data set

```
>par(mfrow=c(1,2))
>pines.G <- Gest(as(spjpines, "ppp"))
>plot(pines.G)
>pines.G
```

##### running the g test by using the 'envelope' code. It will compute Monte Carlo (MC) envelopes of certain types functions. It randomly simulates point patterns and computes G(d) for each point pattern, and then computes Monte Carlo envelopes.

```
>envPines <- envelope(as(spjpines, "ppp"), fun=Gest, nrank=2, nsim=99)
>plot(envPines)
```

### reset the plot region

```
>par(mfrow=c(1,1))
```

### now let's run MC envelopes for all three data sets

```
>d <- seq(0,1,0.01)
>envjap <- envelope(as(spjpines1, "ppp"), fun=Gest, r=d, nrank=2, nsim=99)
>envred <- envelope(as(spred, "ppp"), fun=Gest, r=d, nrank=2, nsim=99)
>envcells <- envelope(as(spcells, "ppp"), fun=Gest, r=d, nrank=2, nsim=99)
```

### now combine them into a single data frame

```
>Gresults <- rbind(envjap, envred, envcells)
>Gresults <- cbind(Gresults, y=rep(c("JAPANESE", "REDWOOD", "CELLS"), each=length(d)))
```

### let's have all three MC envelopes next to each other

```
>print(xyplot(obs~theo|y, data=Gresults, type="l",
             xlab = "theoretical", ylab = "observed", # EJP
             panel=function(x, y, subscripts) {
```

```

lpolygon(c(x, rev(x)),
        c(Gresults$lo[subscripts], rev(Gresults$hi[subscripts])),
        border="gray", col="gray"
)
llines(x, y, col="black", lwd=2)
}))

```

##### F function #####

#### the process is same the G function but look at F

### let's try it first for japanese pines data set

```

>par(mfrow=c(1,2))
>pines.F <- Fest(as(spjpines, "ppp"))
>plot(pines.F)
>pines.F

```

### repeat the MC envelope for F for the apanese pines data set

```

>envPines <- envelope(as(spjpines, "ppp"), fun=Fest, nrank=2, nsim=99)
>plot(envPines)

```

#### compute F for all three data set

```

>d <- seq(0,1,0.001)
>Fenvjap<-envelope(as(spjpines1, "ppp"), fun=Fest, r=d, nrank=2, nsim=99)
>Fenvred<-envelope(as(spred, "ppp"), fun=Fest, r=d, nrank=2, nsim=99)
>Fenvcells<-envelope(as(spcells, "ppp"), fun=Fest, r=d, nrank=2, nsim=99)

```

### combine them into a single data frame

```

>Fresults<-rbind(Fenvjap, Fenvred, Fenvcells)
>Fresults<-cbind(Fresults, y=rep(c("JAPANESE", "REDWOOD", "CELLS"), each=length(d)))

```

### let's plot the three data sets MC envelopes

```

>print(xyplot(obs~theo|y, data=Fresults, type="l",
            xlab = "theoretical", ylab = "observed", # EJP
            panel=function(x, y, subscripts) {

```

```

lpolygon(c(x, rev(x)),
         c(Fresults$lo[subscripts], rev(Fresults$hi[subscripts])),
         border="gray", col="gray"
)
llines(x, y, col="black", lwd=2)
}))

```

##### K function #####

#### the process is same the G and F function but look at K

### we will run it first with the cells data set

```

>par(mfrow=c(1,2))
>cells.K <- Kest(as(spcells, "ppp"))
>plot(cells.K)
>cells.K

```

### repeat the MC envelope for K for the cells data set

```

>envCells <- envelope(as(spcells, "ppp"), fun=Kest, nrank=2, nsim=99)
>plot(envCells)

```

#### compute K for all three data set

```

>r<- seq(0,sqrt(2)/6, by=0.005)
>Kenvjap<-envelope(as(sppines1, "ppp"), fun=Kest, r=r, nrank=2, nsim=99)
>Kenvred<-envelope(as(spred, "ppp"), fun=Kest, r=r, nrank=2, nsim=99)
>Kenvcells<-envelope(as(spcells, "ppp"), fun=Kest, r=r, nrank=2, nsim=99)

```

### combine them into a single data frame

```

>Kresults<-rbind(Kenvjap, Kenvred, Kenvcells)
>Kresults<-cbind(Kresults, y=rep(c("JAPANESE", "REDWOOD", "CELLS"), each=length®))

```

### let's plot the three data sets MC envelopes

```

>print(xyplot((obs-theo)~r|y, data=Kresults, type="l",
             ylim= c(-.06, .06), ylab=expression(hat(K) (r) - pi * r^2),
             panel=function(x, y, subscripts) {

```

```
Ktheo<- Kresults$theo[subscripts]
lpolygon(c(r, rev(r)),
         c(Kresults$lo[subscripts]-Ktheo, rev(Kresults$hi[subscripts]-Ktheo)),
         border="gray", col="gray"
)
llines(r, Kresults$obs[subscripts]-Ktheo, lty=2, lwd=1.5, col="black")
}))
```