

Lab 2 Pattern Analysis-First Order Effect

Dominic LaRoche

September 16, 2014

For this lab we use the Japanese black pine sapling data which is dscribed below.

```
rm(list=ls())
library(spatstat)

##
## spatstat 1.38-1      (nickname: 'Le Hardi')
## For an introduction to spatstat, type 'beginner'

data(japanesepines)
jp<-japanesepines
class(jp)

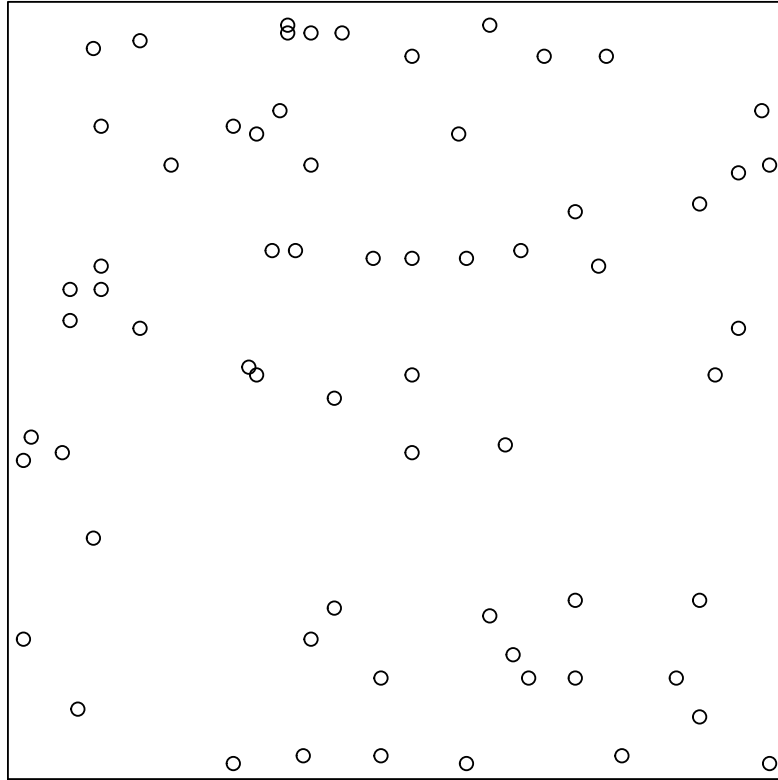
## [1] "ppp"
```

```
summary(jp)

## Planar point pattern: 65 points
## Average intensity 65 points per square unit (one unit = 5.7 metres)
##
## Coordinates are given to 2 decimal places
## i.e. rounded to the nearest multiple of 0.01 units (one unit = 5.7 metres)
##
## Window: rectangle = [0, 1] x [0, 1] units
## Window area = 1 square unit
## Unit of length: 5.7 metres

par(xpd=T)
plot(jp,axes=T, main="Japanese black pine saplings")
```

Japanese black pine saplings



1 Assignment I: Quadrat Tests

We will test for a first order effect using two different quadrat tests, one with 4 cells (2x2) and one with 16 (4x4). The two grids are shown in figure 1.

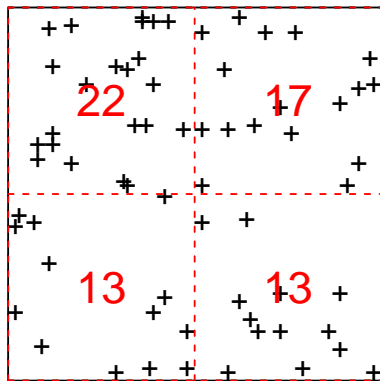
We cannot reject the null hypothesis of complete spatial randomness with either the 2x2 or 4x4 quadrat tests ($\alpha \leq 0.1$). However, the test statistic was highly dependent on the number of quadrats selected (the 3x3 test was nearly significant), which is not a desirable property. Moreover, Figures 2 and 3 show the counts and expected values for each cell.

```

par(mfrow=c(1,2))
#2x2
q.jp2<-quadratcount(jp,nx=2,ny=2)
plot (jp, pch="+", main = " 2x2 Quadrat for jp data")
plot (q.jp2, add=TRUE, col="red", cex=1.5, lty=2)
#4x4
q.jp4<-quadratcount(jp,nx=4,ny=4)
plot (jp, pch="+", main = "4x4 Quadrat for jp data")
plot (q.jp4, add=TRUE, col="red", cex=1.5, lty=2)

```

2x2 Quadrat for jp data



4x4 Quadrat for jp data

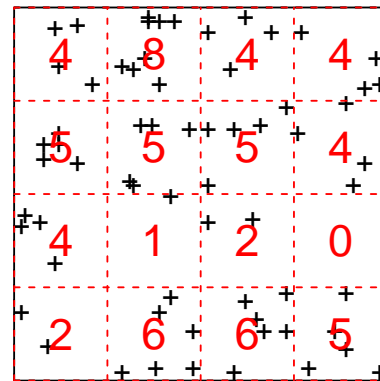


Figure 1: Quadrat counts for the Japanese black pine data.

```

qt.jp2<-quadrat.test(jp,2,2)
qt.jp2

##
## Chi-squared test of CSR using quadrat counts
## Pearson X2 statistic
##
## data:  jp
## X2 = 3.369, df = 3, p-value = 0.6762
## alternative hypothesis: two.sided
##
## Quadrats: 2 by 2 grid of tiles

qt.jp4<-quadrat.test(jp,4,4)

## Warning:  Some expected counts are small; chi^2 approximation may be inaccurate

qt.jp4

##
## Chi-squared test of CSR using quadrat counts
## Pearson X2 statistic
##
## data:  jp
## X2 = 15, df = 15, p-value = 0.9028
## alternative hypothesis: two.sided
##
## Quadrats: 4 by 4 grid of tiles

```

```
plot(qt.jp2, main="Quadrat Test: 2x2 JP data")
plot(jp, pch="+", col="red", add=T)
```

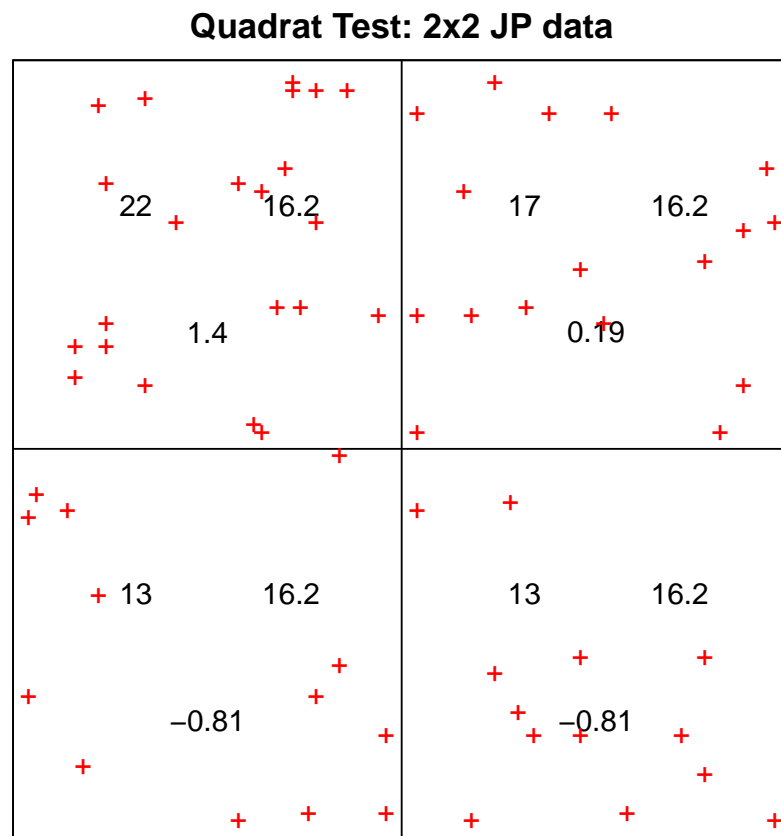


Figure 2: A plot of the test for the 2x2 quadrat test.

```
plot(qt.jp4, main="Quadrat Test: 4x4 JP data")
plot(jp, pch="+", col="red", add=T)
```

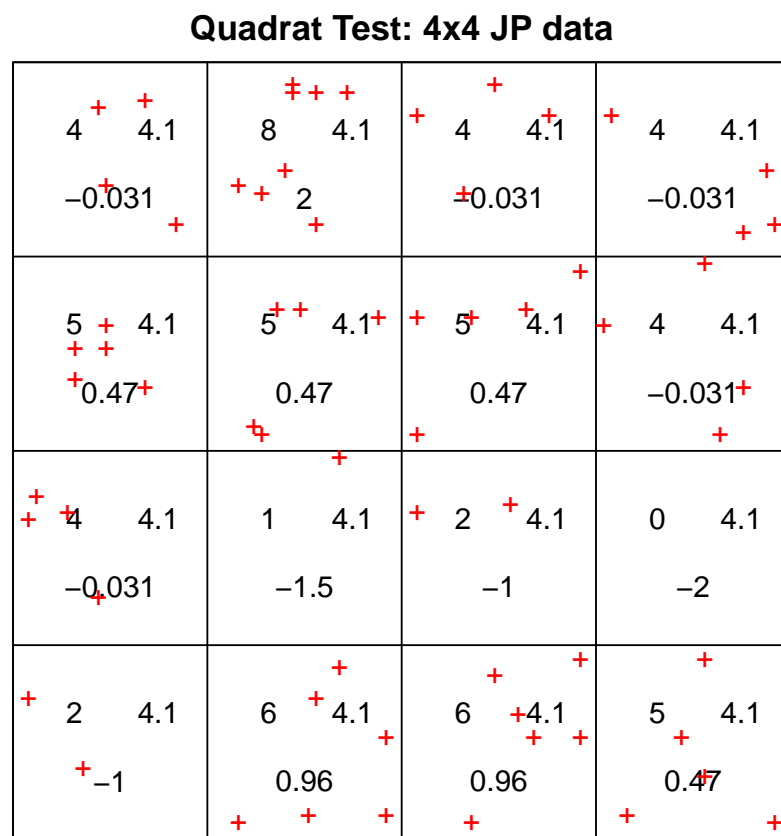


Figure 3: A plot of the test for the 4x4 quadrat test.

2 Assignment II: Kernel estimation

Kernel estimation is subject to the bandwidth selected by the user. Figure 4 shows two different realizations of kernel density estimation with bandwidths equal to 0.05 (top) and 0.1 (bottom). The smaller bandwidth gives a much closer fit to the actual points but this may not approximate the underlying process very well. The larger bandwidth may over smooth the underlying density and hide real patterns.

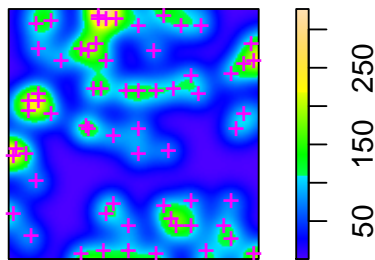
```

# ?density.ppp
par(mfrow=c(2,1))
jp.Z.0.5<-density.ppp(jp, 0.05)
plot(jp.Z.0.5, main="Kernel Estimation of JP: sigma=0.05")
points(jp, pch="+", col="6")

jp.Z.1<-density.ppp(jp, 0.1)
plot(jp.Z.1, main="Kernel Estimation of JP: sigma =0.1")
points(jp, pch="+", col="6")

```

Kernel Estimation of JP : sigma=0.05



Kernel Estimation of JP : sigma =0.1

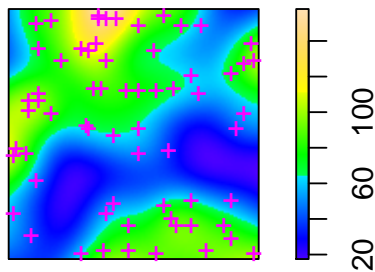
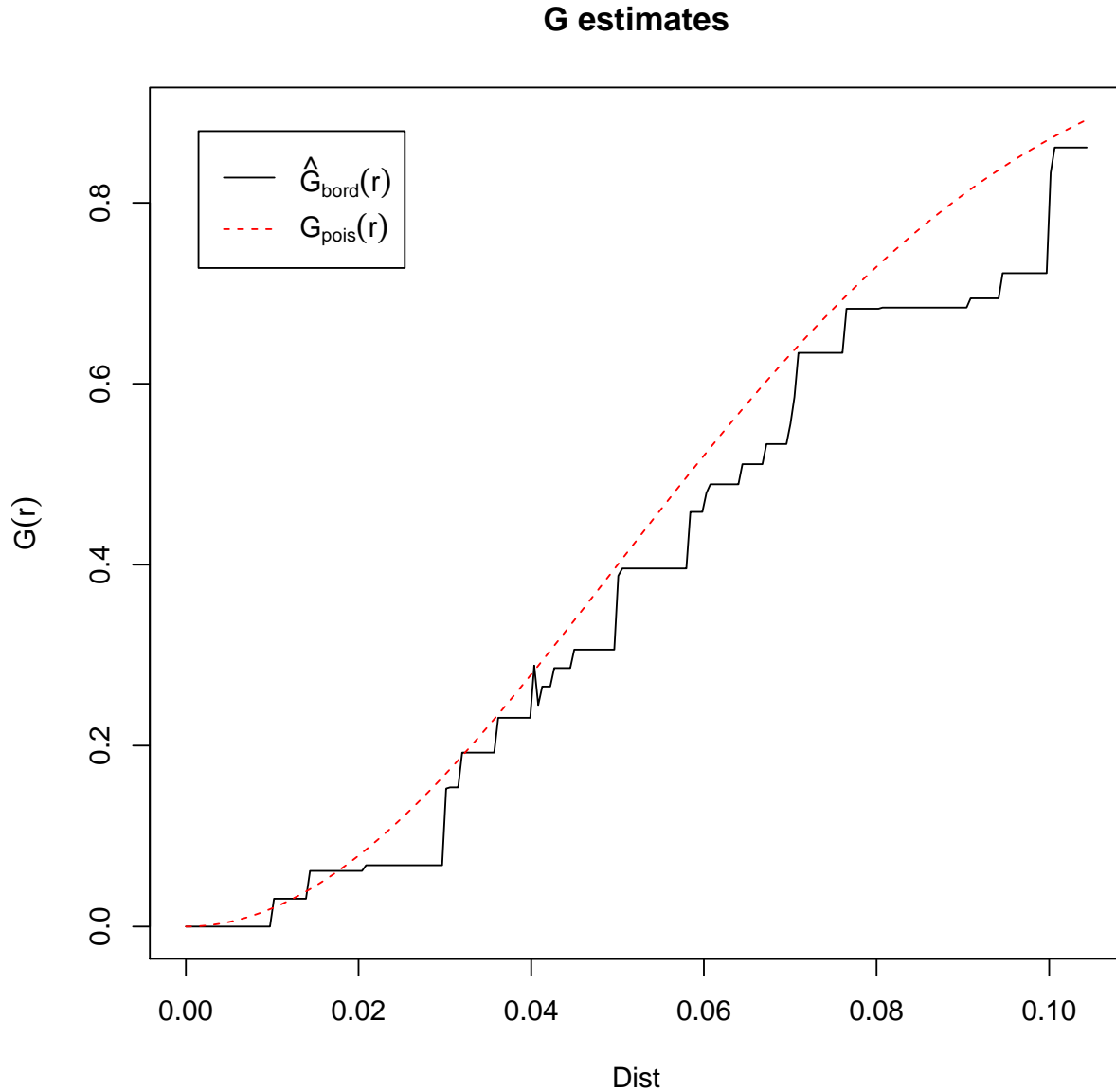


Figure 4: Two different kernel density estimation with sigma =0.05 and 0.1.

3 Assignemnt 3: Second Order Effect

The plots of \hat{G} (fig. 5) and \hat{F} (fig. 6) both suggest that the pattern is random since there are no serious deviations between the theoretical and observed distances. The \hat{K} function plot (fig. 7) and the \hat{L} function both suggest slight evidence of a regular pattern but the significance of this cannot be ascertained from the plots without confidence bands.

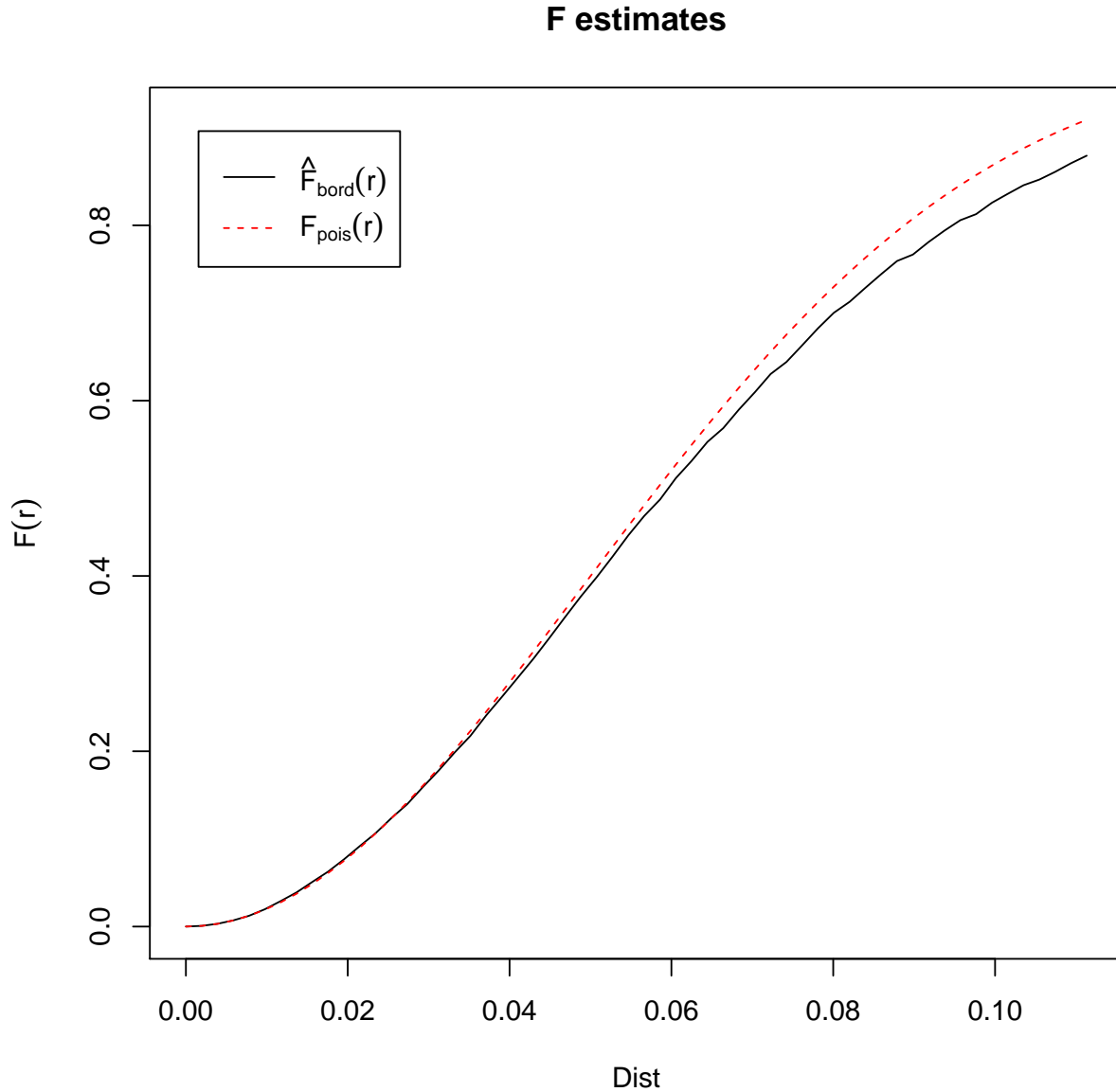
```
jp.ghat<-Gest(jp)
g.max<-max(jp.ghat$r)
plot(jp.ghat, cbind(rs, theo)~r, main="G estimates", xlab="Dist")
```



##	lty	col	key	label	meaning
## rs	1	1	rs	$\hat{G}_{\text{bord}}(r)$	border corrected estimate of $G(r)$
## theo	2	2	theo	$G_{\text{pois}}(r)$	theoretical Poisson $G(r)$

Figure 5: Estimation of the G function using nearest neighbor distances and comparison with a theoretical poisson distribution.

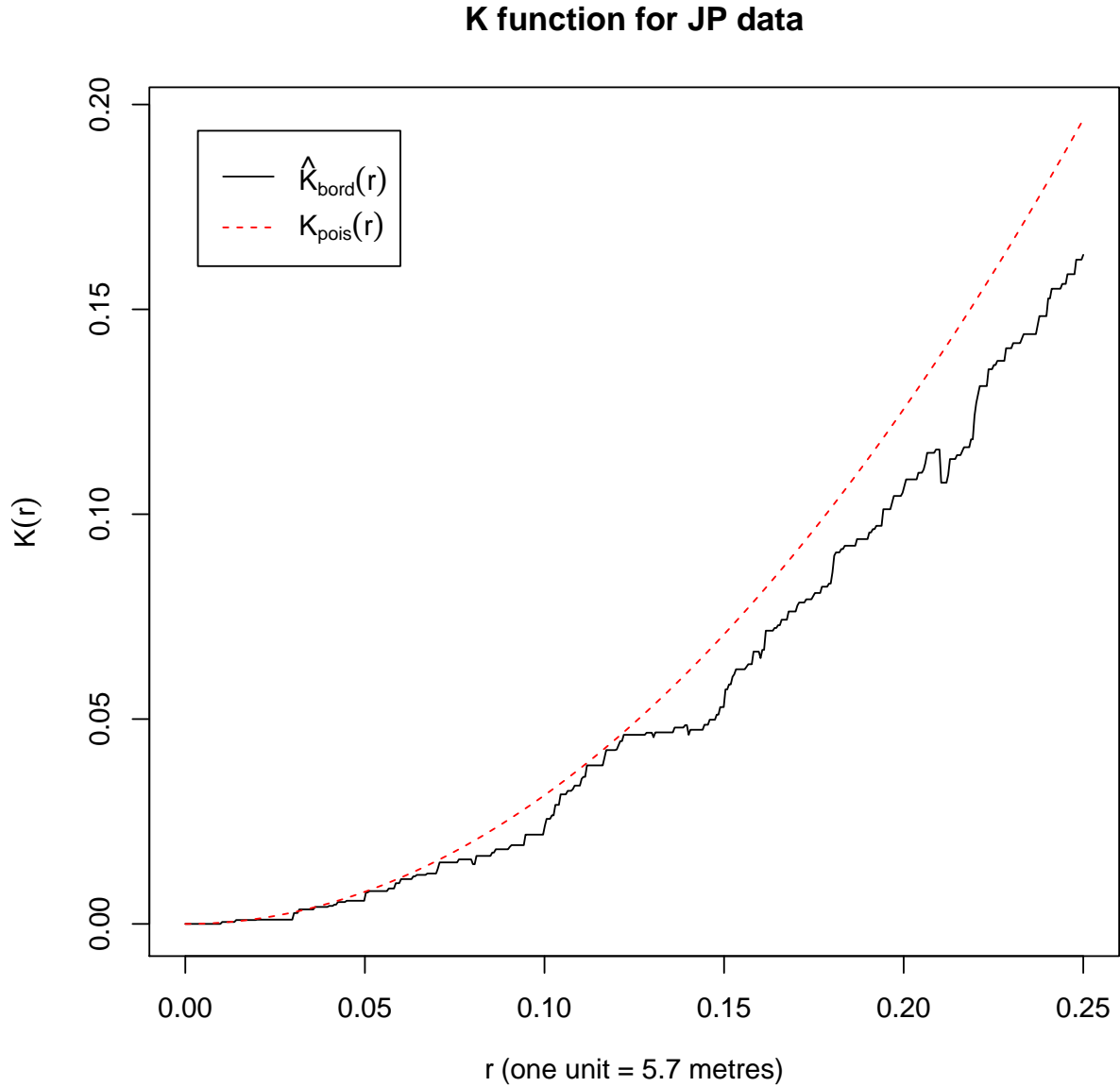
```
jp.fhat<-Fest(jp)
f.max<-max(jp.fhat$r)
plot(jp.fhat, cbind(rs, theo)~r, main="F estimates", xlab="Dist")
```



##	lty	col	key	label	meaning
## rs	1	1	rs	$\hat{F}_{\text{bord}}(r)$	border corrected estimate of $F(r)$
## theo	2	2	theo	$F_{\text{pois}}(r)$	theoretical Poisson $F(r)$

Figure 6: Estimation of the G function using point-to-event distances and comparison with a theoretical poisson distribution.

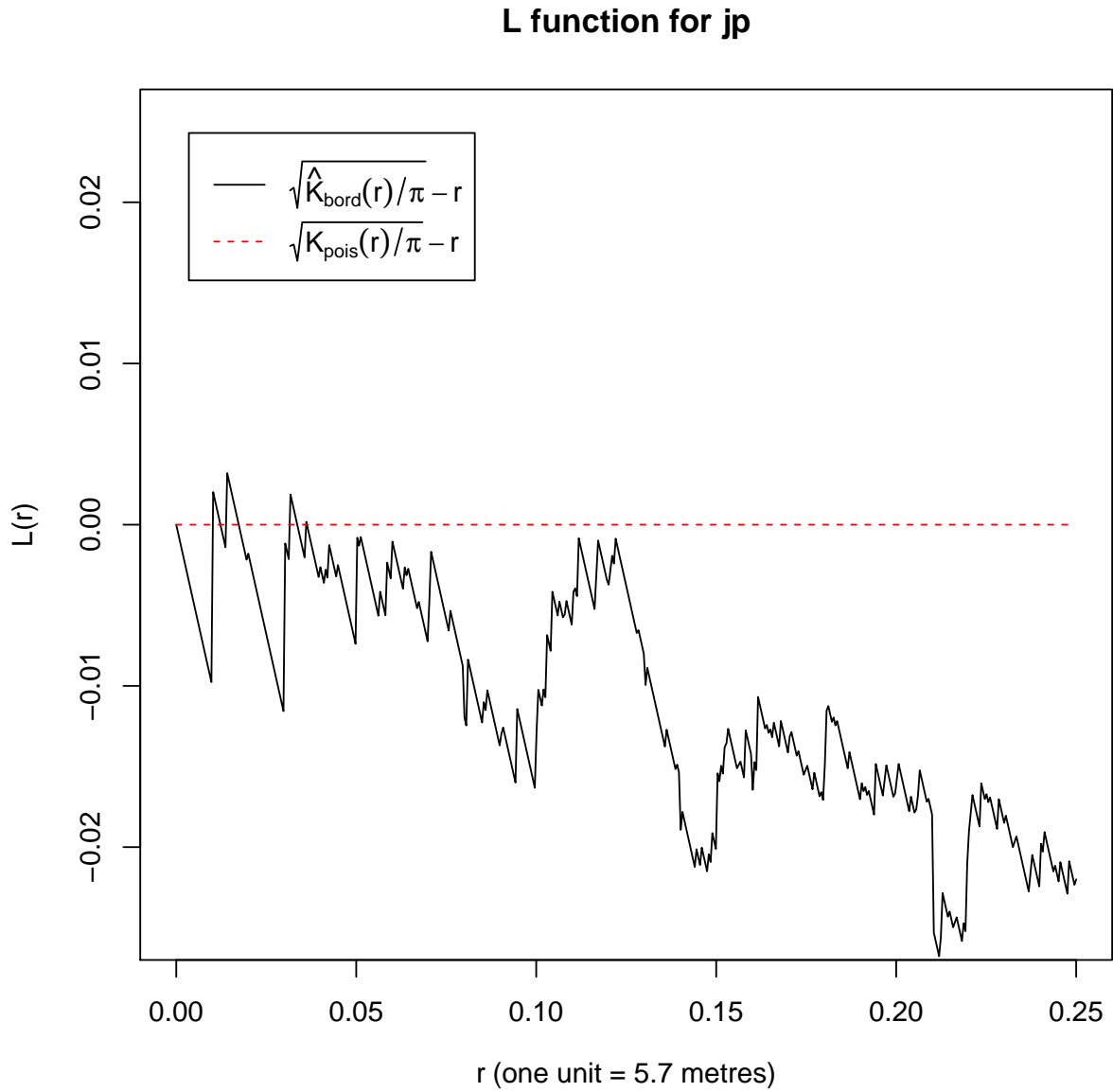
```
jp.khat<-Kest(jp)
plot(jp.khat, cbind(border, theo)~r, main="K function for JP data")
```



##	lty	col	key	label	meaning
## border	1	1	border	$\hat{K}[\text{bord}](r)$	border-corrected estimate of $K(r)$
## theo	2	2	theo	$K[\text{pois}](r)$	theoretical Poisson $K(r)$

Figure 7: Plot of the K function for the japanese pine data showing the same pattern as the \hat{G} and \hat{F} plots.

```
plot(jp.khat, sqrt(cbind(border, theo)/pi)-r~r,ylab="L(r)", main="L function for jp", ylim=c(-0.025, 0.025))
```



```
##      lty col  key      label
## border  1  1 border sqrt(hat(K)[bord](r)/pi)-r
## theo    2  2 theo    sqrt(K[pois](r)/pi)-r
##                                     meaning
## border border-corrected estimate of K(r)
## theo    theoretical Poisson K(r)
```

Figure 8: The \hat{L} plot of the same japaneses pine data suggesting a regular pattern in the data.