test_pkg

2017-07-21

Contents

```
library(mytestpkg)
```

Use new functions:

1.

```
a <- c(5,3,9,5)
b <- c(4,6,7,3)
new_fun(a,b)
```

```
## [1] -3.0 -5.4 -5.2 -2.0
```

Create Table **??** (after SCHAUER and FAILLARD 1968):

Create Plot 1 :

```
plot(tab)
```

For help check Hijmans (2016)

```
library(binford)
data(LRB)
knitr::kable(head(LRB))
```

X	seq339	groupno	name	year	ethref
Punan	1	1	Punan_(Borneo)	1970	Kedit 1982 Harrison 1949 Avadhani 1975
Batek	2	2	Batek_Phillipines	1968	Eder 1987 Cadelina 1982
Kubu	3	3	Kubu-(Ridan)	1900	NA
Shompen	4	4	Shompen	1989	Rivzi 1990
Onge	5	5	Onge	1952	Heine-Geldern Hoehn-Gerlachstein 1958 Sen 1962 Cooper 1991
Jarwa	6	6	Jarwa	1906	Temple 1903 Radcliffe-Brown 1948

1 Point Pattern

```
##------------------------
## First Order effects
##------------------------
```

Table 1: My new table

	a	b
test1	3	5
test2	5	3
test3	8	6
test4	3	4

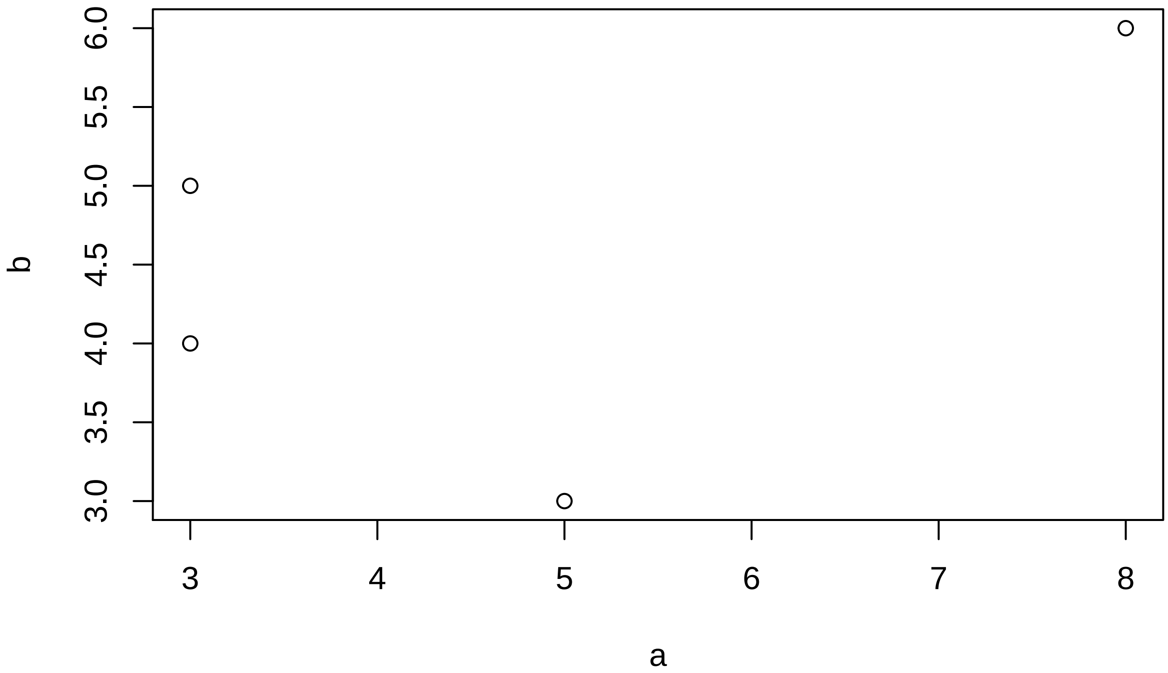


Figure 1: Plot Table

```
harran <- read.table("../data/ReReLA/data/Sites_HarranPlain.csv", sep=",", header=TRUE)
head(harran)
```

```
##   X.1         Name        X        Y           Mentioned_Epoch
## 1   1   Anep Sırtı 38.76667 37.63333         Upper Palaeolithic
## 2   2  Çekiş Sırtı 38.85000 37.66667 Middle Palaeolithic Period
## 3   3  Çekiş Sırtı 38.85000 37.66667          Lower Paleolithic
## 4   4 Göbekli Tepe 38.92247 37.22328                       PPNA
## 5   5 Göbekli Tepe 38.92247 37.22328                      EPPNB
## 6   6 Göbekli Tepe 38.92247 37.22328                      MPPNB
```

```
## Create Spatial Object
##--------------------------
library(sp)
coordinates(harran) <- ~X+Y
proj4string(harran) <- CRS("+init=epsg:4326")
harran <- spTransform(harran, CRSobj = CRS("+init=epsg:32637"))

## Create Point pattern object
##------------------------------
library(spatstat)
```

```
## Loading required package: nlme
```

```
##
## Attaching package: 'nlme'
```

```
## The following object is masked from 'package:raster':
##
##     getData
```

```
## Loading required package: rpart
```

```
##
## spatstat 1.51-0       (nickname: 'Poetic Licence')
## For an introduction to spatstat, type 'beginner'

##
## Note: spatstat version 1.51-0 is out of date by more than 11 weeks; a newer version should be availa

##
## Attaching package: 'spatstat'

## The following objects are masked from 'package:raster':
##
##     area, rotate, shift
```

```
harran_ppp <- ppp(x = harran@coords[,1],
                  y = harran@coords[,2],
                  window = owin(xrange = harran@bbox[1,],
                                yrange = c(min(harran@bbox[2,]), min(harran@coords[,2]+52000)))
                  )
```

```
## Warning: 65 points were rejected as lying outside the specified window

## Warning: data contain duplicated points
```

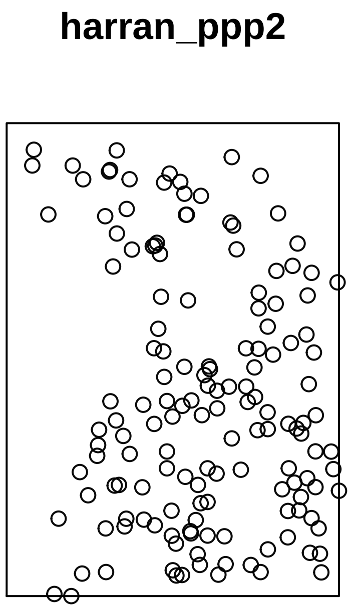
```
# remove duplicated points
str(harran_ppp)
```

```
## List of 5
##  $ window    :List of 4
##   ..$ type  : chr "rectangle"
##   ..$ xrange: Named num [1:2] 477942 514430
##   .. ..- attr(*, "names")= chr [1:2] "min" "max"
##   ..$ yrange: num [1:2] 4062337 4114337
##   ..$ units :List of 3
##   .. ..$ singular  : chr "unit"
##   .. ..$ plural    : chr "units"
##   .. ..$ multiplier: num 1
##   .. ..- attr(*, "class")= chr "units"
##   ..- attr(*, "class")= chr "owin"
##  $ n         : int 279
##  $ x         : num [1:279] 485197 485197 491077 491077 491077 ...
##  $ y         : num [1:279] 4109677 4109677 4070842 4070842 4070842 ...
##  $ markformat: chr "none"
##  - attr(*, "class")= chr "ppp"
##  - attr(*, "rejects")=List of 5
##   ..$ window    :List of 5
##   .. ..$ type  : chr "polygonal"
##   .. ..$ xrange: num [1:2] 477664 514695
##   .. ..$ yrange: num [1:2] 4060911 4292855
##   .. ..$ bdry  :List of 1
##   .. .. ..$ :List of 2
##   .. .. .. ..$ x: num [1:10] 505940 502993 495642 477664 479054 ...
##   .. .. .. ..$ y: num [1:10] 4275964 4281593 4292855 4165265 4123978 ...
##   .. ..$ units :List of 3
##   .. .. ..$ singular  : chr "unit"
##   .. .. ..$ plural    : chr "units"
##   .. .. ..$ multiplier: num 1
##   .. .. ..- attr(*, "class")= chr "units"
##   .. ..- attr(*, "class")= chr "owin"
##   ..$ n         : int 65
##   ..$ x         : num [1:65] 479412 486771 486771 493122 493122 ...
##   ..$ y         : num [1:65] 4165159 4168843 4168843 4119645 4119645 ...
##   ..$ markformat: chr "none"
##   ..- attr(*, "class")= chr "ppp"
```

```
harran_ppp2 <- harran_ppp[!duplicated(harran_ppp)]
str(harran_ppp2)
```

```
## List of 5
##  $ window    :List of 4
##   ..$ type  : chr "rectangle"
##   ..$ xrange: Named num [1:2] 477942 514430
##   .. ..- attr(*, "names")= chr [1:2] "min" "max"
##   ..$ yrange: num [1:2] 4062337 4114337
##   ..$ units :List of 3
##   .. ..$ singular  : chr "unit"
##   .. ..$ plural    : chr "units"
##   .. ..$ multiplier: num 1
##   .. ..- attr(*, "class")= chr "units"
##   ..- attr(*, "class")= chr "owin"
##  $ n         : int 149
##  $ x         : num [1:149] 485197 491077 482518 497239 495545 ...
##  $ y         : num [1:149] 4109677 4070842 4104300 4083259 4083780 ...
##  $ markformat: chr "none"
##  - attr(*, "class")= chr "ppp"
```

```
plot(harran_ppp2)
```



```
# or:
anyDuplicated(harran_ppp)
```

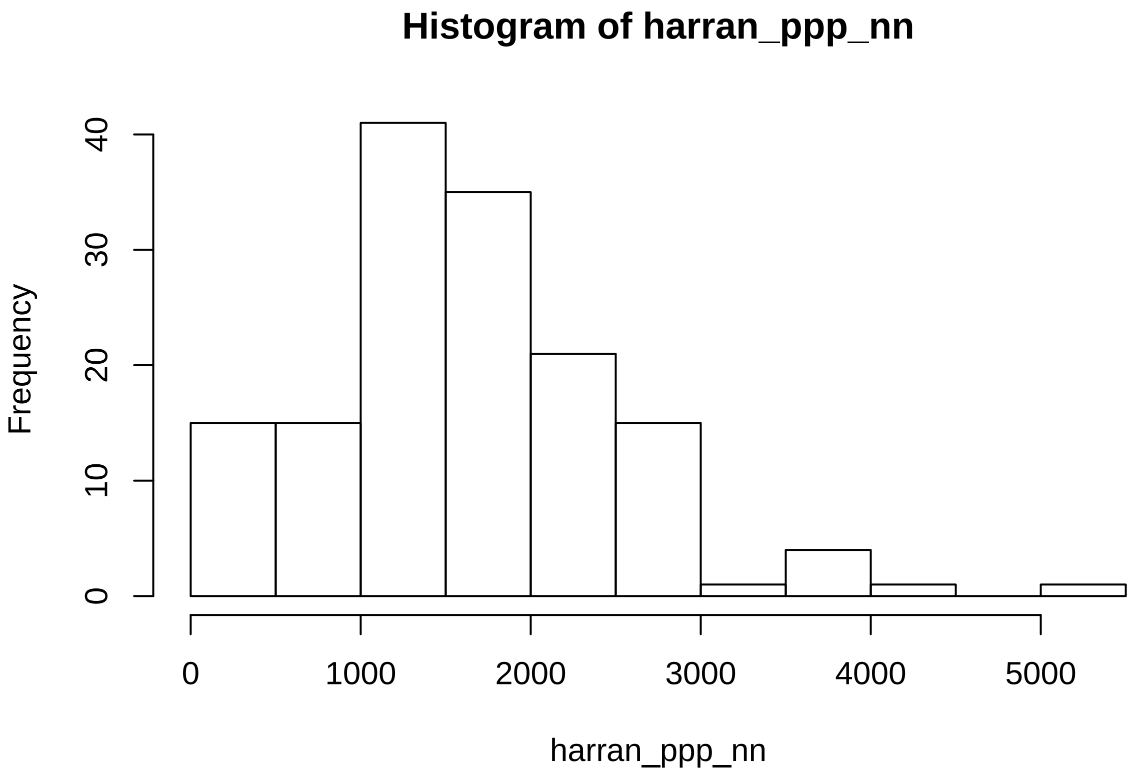
```
## [1] 2
```

```
harran_ppp3 <- unique(harran_ppp)
```

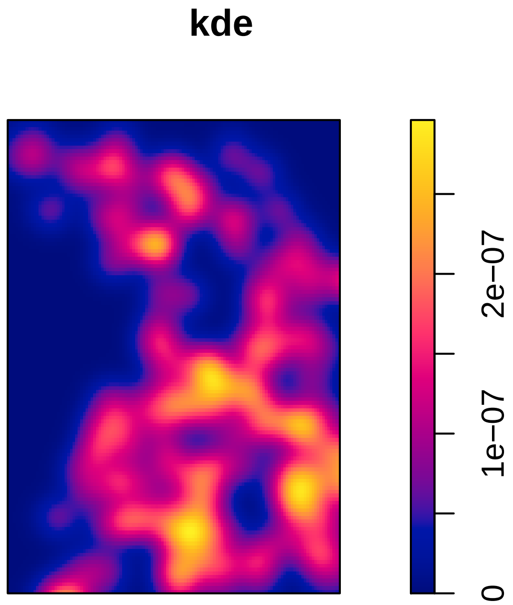
```r
harran_ppp_nn <- nndist(harran_ppp2)
str(harran_ppp_nn)
```

```
##  num [1:149] 1896 868 5436 1149 1772 ...
```

```r
hist(harran_ppp_nn)
```



```r
## Kernel Density Estimation
##-----------------------------
kde <- density.ppp(x=harran_ppp2, sigma = mean(harran_ppp_nn))
plot(kde)
```



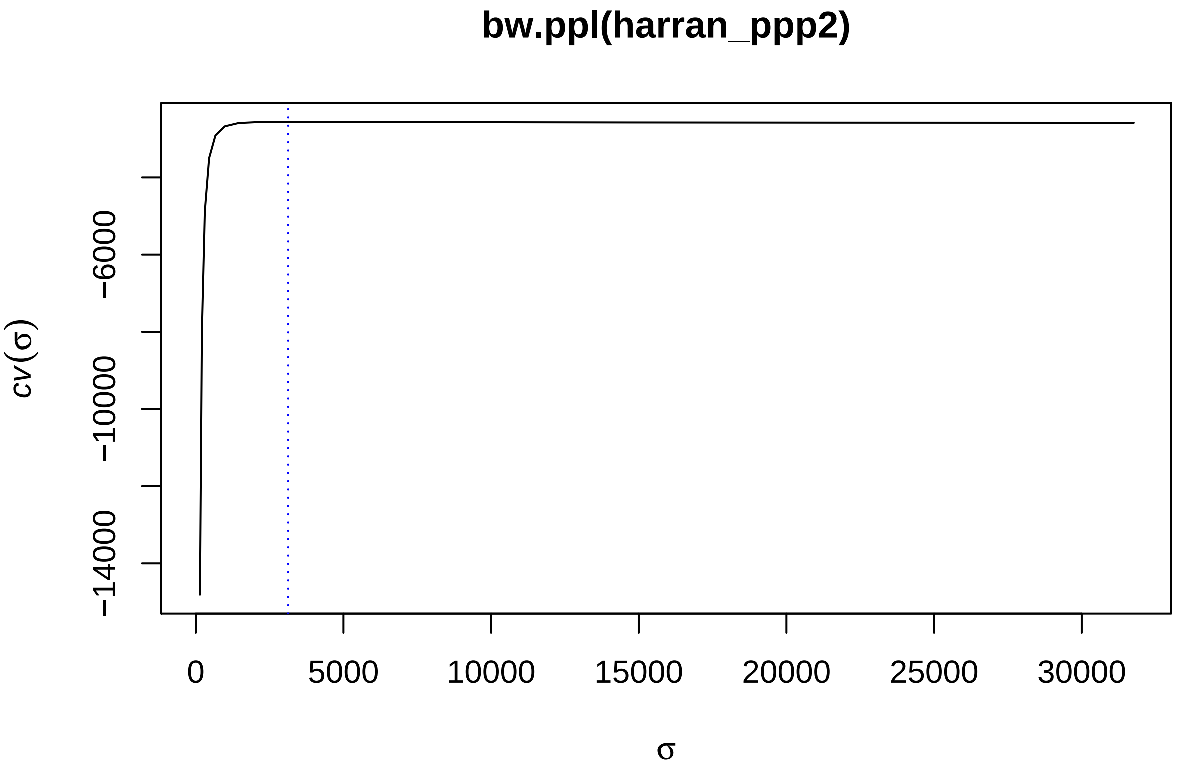
```
# use another bandwidth (sigma)
# for clustered data use diggle
bw.ppl(harran_ppp2)
```

```
##    sigma
## 3126.158
```

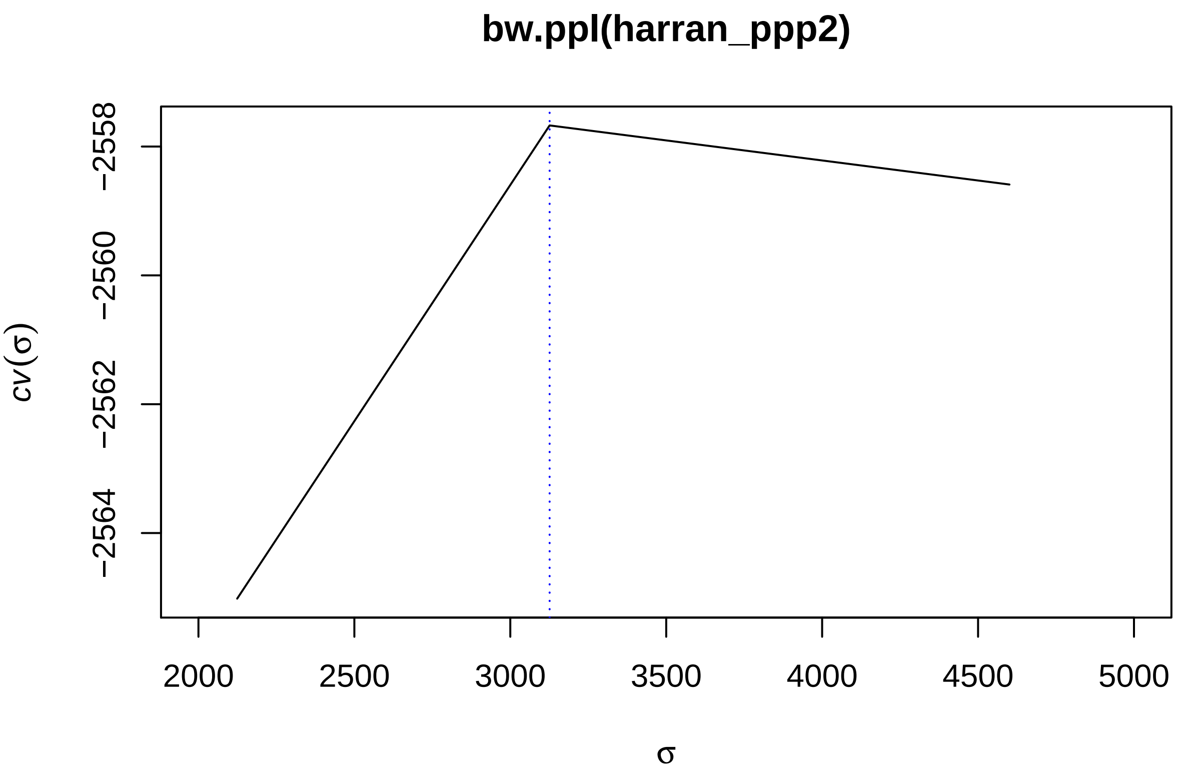
```
bw.diggle(harran_ppp2)
```

```
##    sigma
## 2231.426
```

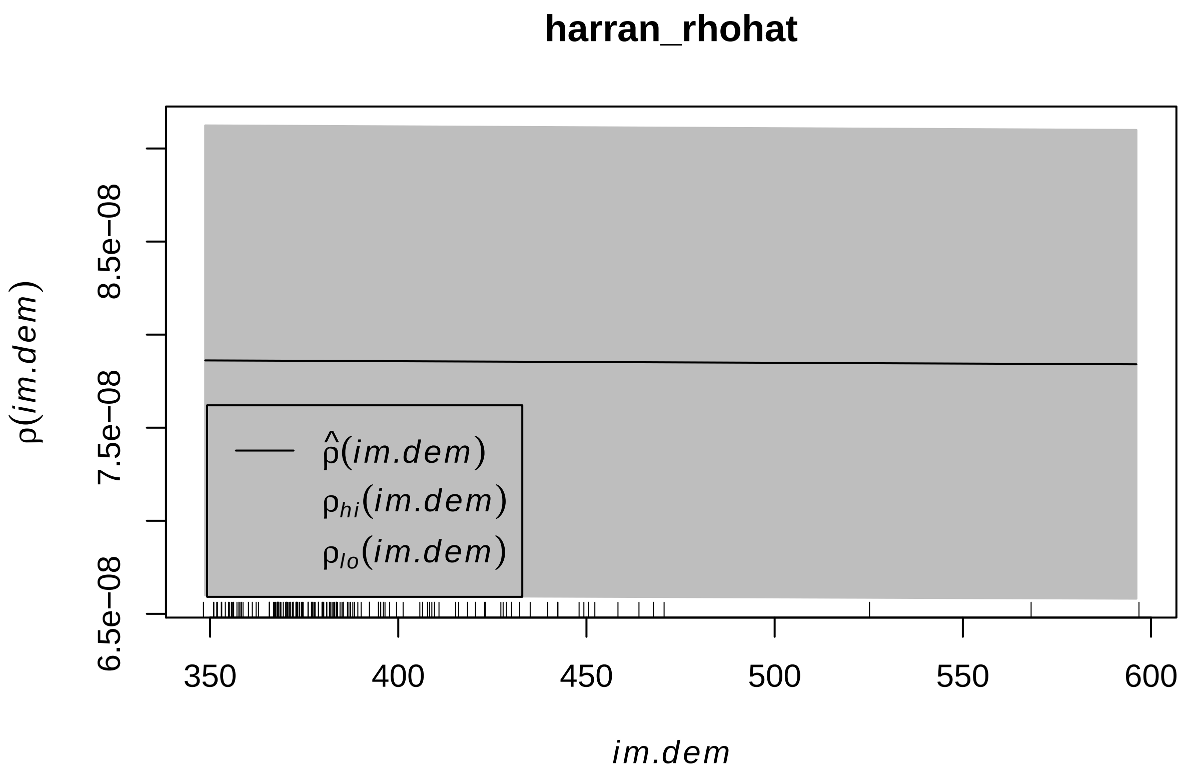
```
plot(bw.ppl(harran_ppp2))
```



```
plot(bw.ppl(harran_ppp2), xlim=c(2000,5000))
```



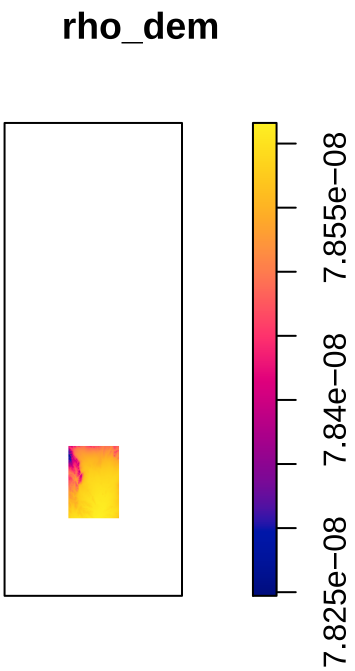
```
## Add a Covariate
##----------------------------
# load raster
library(raster)
dem <- raster("../data/ReReLA/data/dem_harran.tif")

# convert raster to pixel image
t <- as(dem, "SpatialGridDataFrame")
im.dem <- as.im(as.image.SpatialGridDataFrame(as(dem, "SpatialGridDataFrame")))
harran_rhohat <-rhohat(object = harran_ppp2, covariate = im.dem, bw = mean(harran_ppp_nn))      # window
plot(harran_rhohat)    # y-axis: intensity of points
```



```r
# check bandwidth (sigma):
str(harran_rhohat)
```

```
## Classes 'rhohat', 'fv' and 'data.frame': 512 obs. of  5 variables:
##  $ im.dem: num  347 348 349 350 350 ...
##  $ rho   : num  7.86e-08 7.86e-08 7.86e-08 7.86e-08 7.86e-08 ...
##  $ var   : num  4.15e-17 4.15e-17 4.15e-17 4.15e-17 4.15e-17 ...
##  $ hi    : num  9.12e-08 9.12e-08 9.12e-08 9.12e-08 9.12e-08 ...
##  $ lo    : num  6.6e-08 6.6e-08 6.6e-08 6.6e-08 6.6e-08 ...
##  - attr(*, "argu")= chr "im.dem"
##  - attr(*, "valu")= chr "rho"
##  - attr(*, "ylab")= language rho(im.dem)
##  - attr(*, "yexp")= language rho(im.dem)
##  - attr(*, "fmla")= chr ".~im.dem"
##  - attr(*, "alim")= num  348 597
##  - attr(*, "labl")= chr  "im.dem" "hat(%s)(im.dem)" "bold(Var)~hat(%s)(im.dem)" "%s[hi](im.dem)" ...
##  - attr(*, "desc")= chr  "covariate im.dem" "Estimated intensity" "Variance of estimator" "Upper lim
##  - attr(*, "units")=List of 3
##   ..$ singular  : chr "unit"
##   ..$ plural    : chr "units"
##   ..$ multiplier: num 1
##   ..- attr(*, "class")= chr "units"
##  - attr(*, "fname")= chr "rho"
##  - attr(*, "dotnames")= chr  "rho" "hi" "lo"
##  - attr(*, "stuff")=List of 11
##   ..$ modelcall : NULL
##   ..$ callstring: chr "rhohat.ppp(object = harran_ppp2, covariate = im.dem, bw = mean(harran_ppp_nn)
##   ..$ sigma     : num 1659
##   ..$ covname   : chr "im.dem"
##   ..$ ZX        : num  464 371 597 372 373 ...
##    ..$ lambda    : num  7.85e-08 7.85e-08 7.85e-08 7.85e-08 7.85e-08 ...
##    ..$ method    : chr "ratio"
##    ..$ smoother  : chr "kernel"
##    ..$ reference : chr "Lebesgue"
##    ..$ horvitz   : logi FALSE
##    ..$ Zimage    :List of 10
##    .. ..$ v     : num [1:1842, 1:871] NA NA NA NA NA NA NA NA NA NA ...
##    .. ..$ dim   : int  1842 871
##    .. ..$ xrange: num  432102 559965
##    .. ..$ yrange: num  4006377 4347147
##    .. ..$ xstep : num 147
##    .. ..$ ystep : num 185
##    .. ..$ xcol  : num  432176 432323 432469 432616 432763 ...
##    .. ..$ yrow  : num  4006469 4006654 4006839 4007024 4007209 ...
##    .. ..$ type  : chr "real"
##    .. ..$ units :List of 3
##    .. .. ..$ singular  : chr "unit"
##    .. .. ..$ plural    : chr "units"
##    .. .. ..$ multiplier: num 1
##    .. .. ..- attr(*, "class")= chr "units"
##    .. ..- attr(*, "class")= chr "im"
```

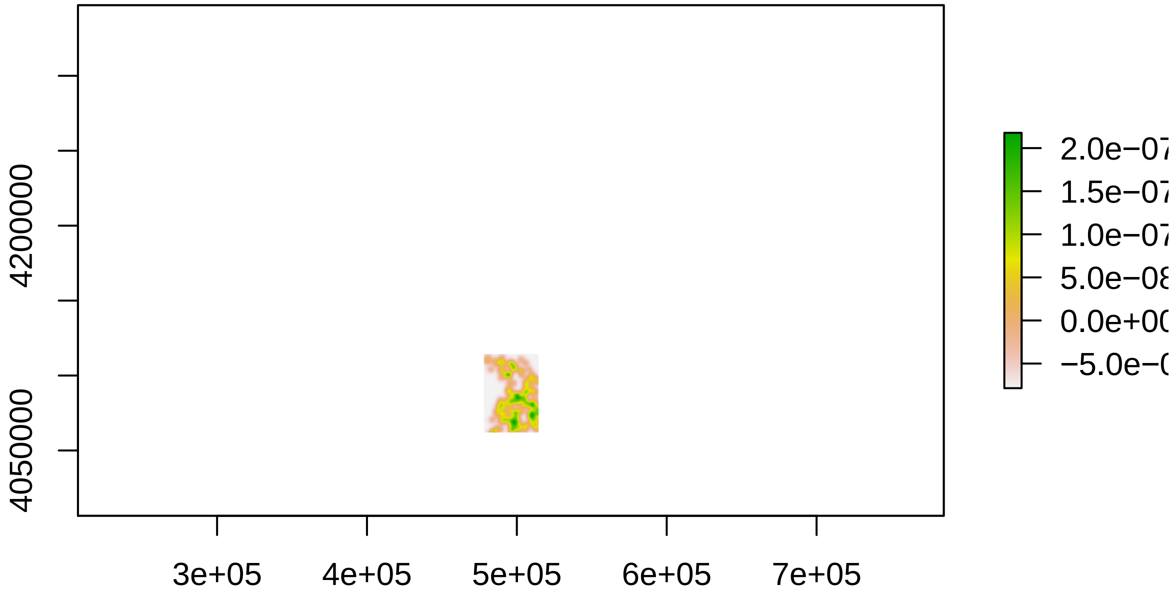
```
# Predict
rho_dem <- predict(harran_rhohat)
plot(rho_dem)
```



```
# Compare raster with real data with predicted raster
diff_rho <- kde - rho_dem
```

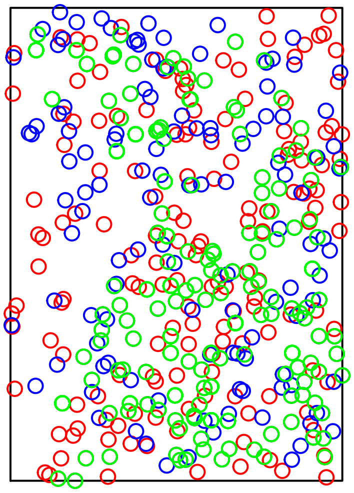
```
## Warning: the images 'e1' and 'e2' were not compatible
```

```
plot(raster(diff_rho))
```



```
## Test against random poisson process
## ------------------------------------
# create random points with the same density like the real points
# compute density - Points per area
dens1 <- harran_ppp2$n/area.owin(harran_ppp2$window)
#set.seed(123)
harran_poispp1 <- rpoispp(lambda = dens1, win = harran_ppp2$window)   # poisson - complete spatial rand

## or /error
dens2 <- intensity(harran_ppp2)
#set.seed(123)
harran_poispp2 <- rpoispp(lambda = dens2, win = harran_ppp2$window)

## or:
#set.seed(123)
harran_poispp3 <- (ex = harran_ppp)

plot(harran_ppp2)
points(harran_poispp1, col="red")
points(harran_poispp2, col="blue")
points(harran_poispp3, col="green")
```

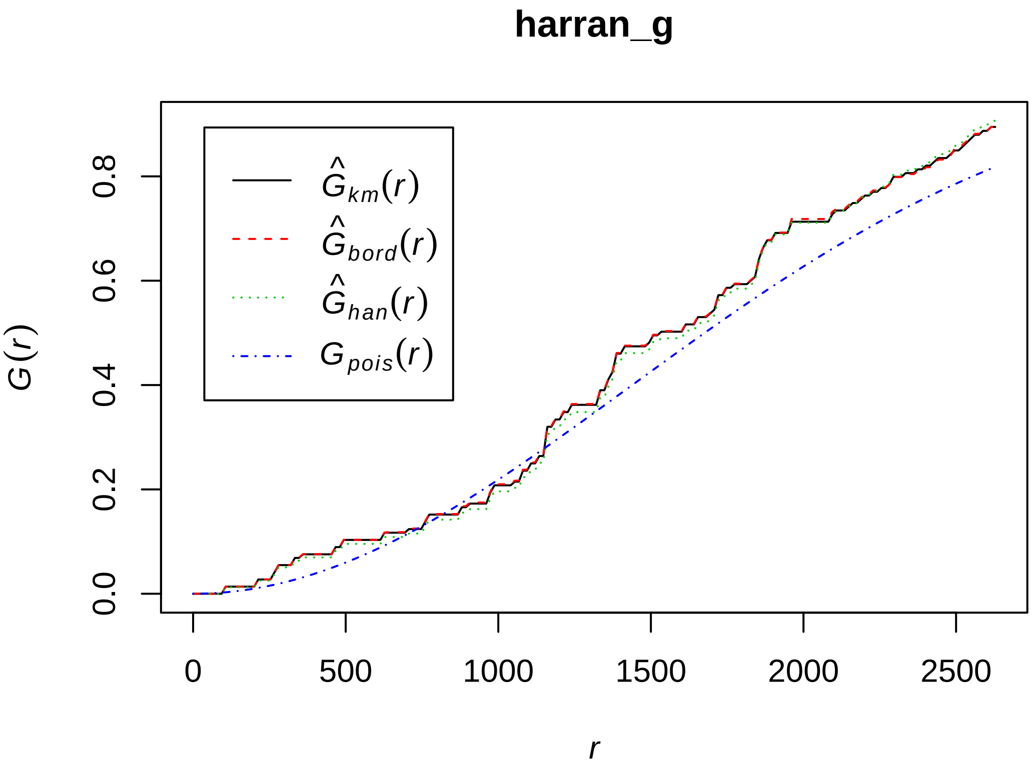
harran_ppp2



```
##------------------------
## Second order effects
##-----------------------
# G-Function
harran_g <- Gest(harran_ppp2)
str(harran_g)
```

```
## Classes 'fv' and 'data.frame':   513 obs. of  7 variables:
##  $ r      : num  0 13.3 26.7 40 53.4 ...
##  $ theo   : num  0.00 4.39e-05 1.76e-04 3.95e-04 7.02e-04 ...
##  $ han    : num  0 0 0 0 0 ...
##  $ rs     : num  0 0 0 0 0 ...
##  $ km     : num  0 0 0 0 0 ...
##  $ hazard : num  0 0 0 0 0 ...
##  $ theohaz: num  0.00 6.58e-06 1.32e-05 1.97e-05 2.63e-05 ...
##  - attr(*, "argu")= chr "r"
##  - attr(*, "valu")= chr "km"
##  - attr(*, "ylab")= language G(r)
##  - attr(*, "yexp")= language G(r)
##  - attr(*, "fmla")= chr ".~r"
##  - attr(*, "alim")= num  0 2628
##  - attr(*, "labl")= chr  "r" "%s[pois](r)" "hat(%s)[han](r)" "hat(%s)[bord](r)" ...
##  - attr(*, "desc")= chr  "distance argument r" "theoretical Poisson %s" "Hanisch estimate of %s" "bo
##  - attr(*, "units")=List of 3
##   ..$ singular  : chr "unit"
##   ..$ plural    : chr "units"
##   ..$ multiplier: num 1
##   ..- attr(*, "class")= chr "units"
##  - attr(*, "fname")= chr "G"
##  - attr(*, "dotnames")= chr  "km" "rs" "han" "theo"
```

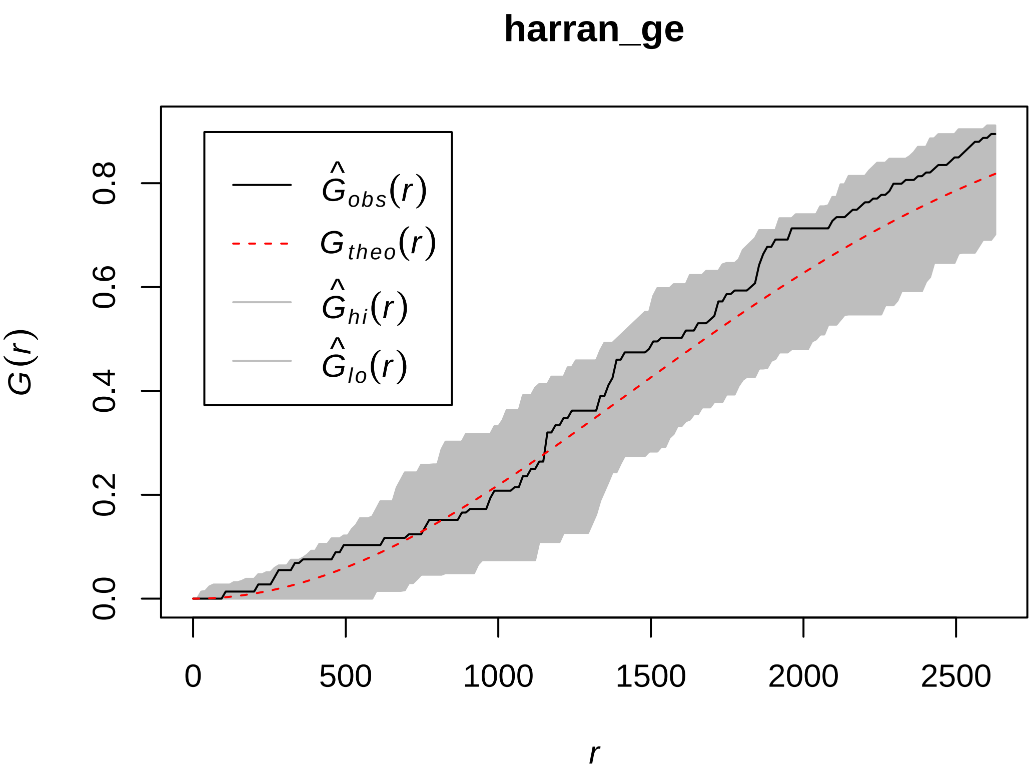
```
plot(harran_g)
```



```r
# generate 99 random points and run G-Function
harran_ge <- envelope(harran_ppp2, fun="Gest")
```

```
## Generating 99 simulations of CSR  ...
## 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28
## 39, 40, 41, 42, 43, 44, 45, 46, 47, 48, 49, 50, 51, 52, 53, 54, 55, 56, 57, 58, 59, 60, 61, 62, 63, 6
## 77, 78, 79, 80, 81, 82, 83, 84, 85, 86, 87, 88, 89, 90, 91, 92, 93, 94, 95, 96, 97, 98,  99.
##
## Done.
```

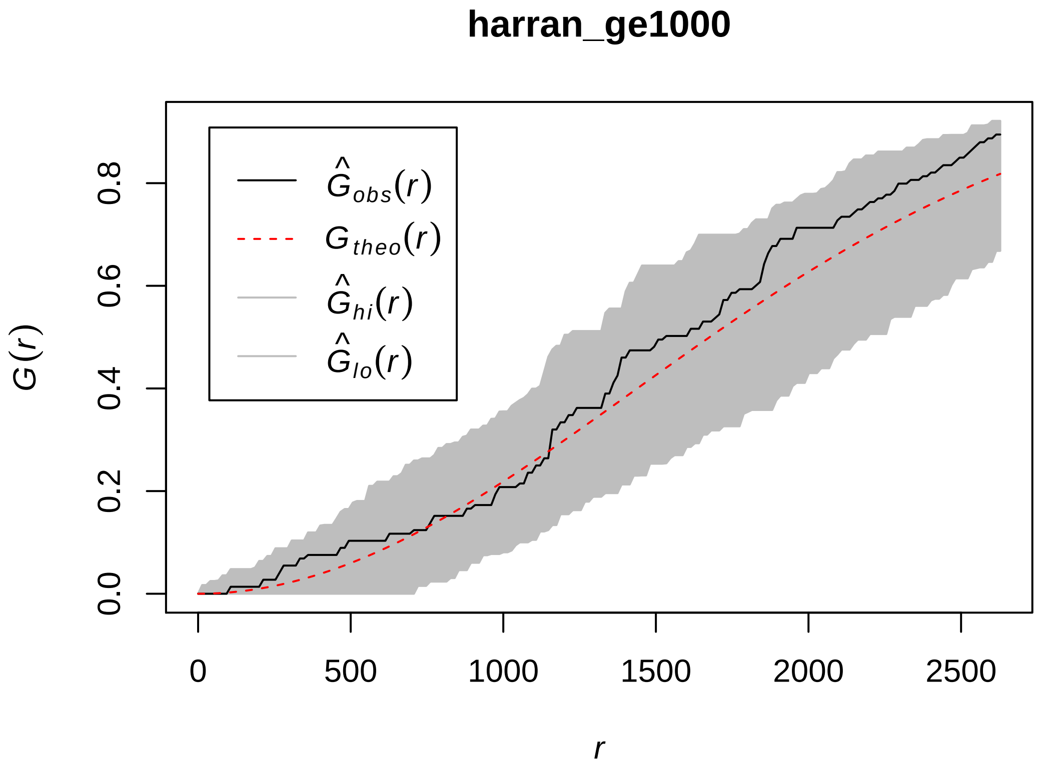
```r
plot(harran_ge)
```



```
# greyish curve - result of all 99 cases - becomes wider with more simulations
harran_ge1000 <- envelope(harran_ppp2, fun="Gest", nsim=1000)
```

```
## Generating 1000 simulations of CSR  ...
## 1, 2, 3, ......10.........20.........30.........40.........50.........60...
## ......70.........80.........90.........100.........110.........120......
## ...130.........140.........150.........160.........170.........180.........
## 190.........200.........210.........220.........230.........240.........250..
## .......260.........270.........280.........290.........300.........310.....
## ....320.........330.........340.........350.........360.........370........
## .380.........390.........400.........410.........420.........430.........440.
## ........450.........460.........470.........480.........490.........500....
## .....510.........520.........530.........540.........550.........560.......
## ..570.........580.........590.........600.........610.........620.........630
## .........640.........650.........660.........670.........680.........690...
## ......700.........710.........720.........730.........740.........750......
## ...760.........770.........780.........790.........800.........810.........
## 820.........830.........840.........850.........860.........870.........880..
## .......890.........900.........910.........920.........930.........940.....
## ....950.........960.........970.........980.........990......... 1000.
##
## Done.
```

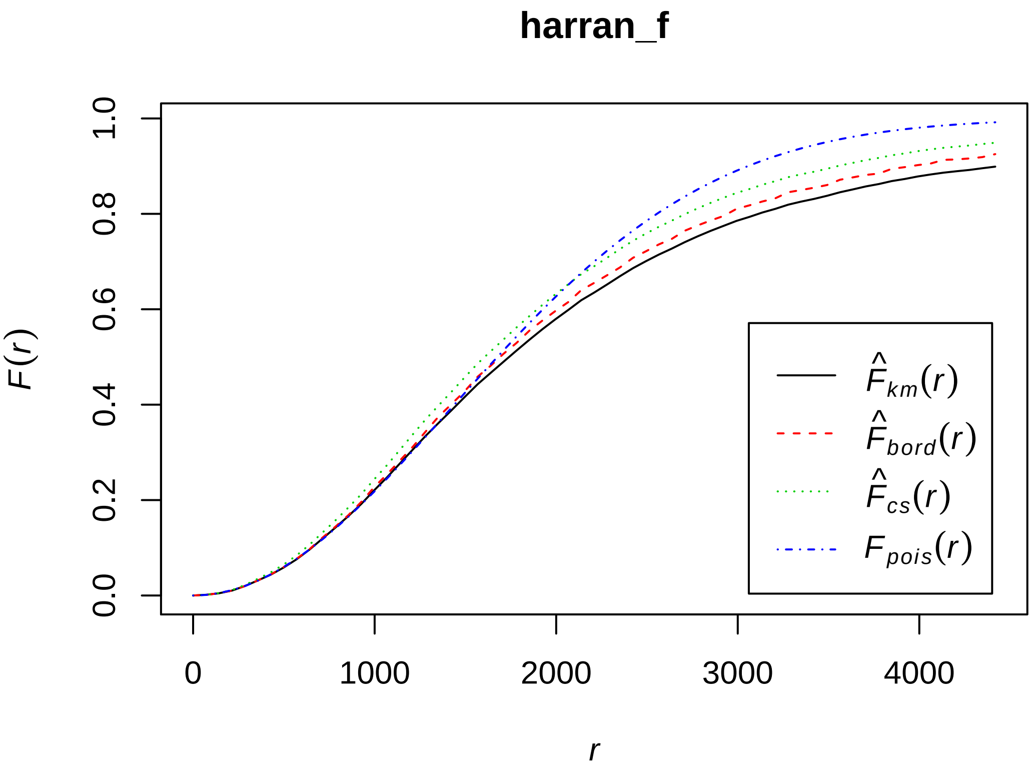
```
plot(harran_ge1000)
```



```r
# F-Function
harran_f <- Fest(harran_ppp2)
str(harran_f)
```

```
## Classes 'fv' and 'data.frame':   97 obs. of  7 variables:
##  $ r      : num  0 71.3 142.5 213.8 285.1 ...
##  $ theo   : num  0 0.00125 0.005 0.01121 0.01985 ...
##  $ cs     : num  0 0.00165 0.00517 0.01135 0.02191 ...
##  $ rs     : num  0 0.00146 0.00472 0.01033 0.01984 ...
##  $ km     : num  0 0.00146 0.00468 0.01028 0.0198 ...
##  $ hazard : num  0.00 2.06e-05 4.52e-05 7.93e-05 1.36e-04 ...
##  $ theohaz: num  0.00 3.52e-05 7.03e-05 1.05e-04 1.41e-04 ...
##  - attr(*, "argu")= chr "r"
##  - attr(*, "valu")= chr "km"
##  - attr(*, "ylab")= language F(r)
##  - attr(*, "yexp")= language F(r)
##  - attr(*, "fmla")= chr ".~r"
##  - attr(*, "alim")= num  0 4419
##  - attr(*, "labl")= chr  "r" "%s[pois](r)" "hat(%s)[cs](r)" "hat(%s)[bord](r)" ...
##  - attr(*, "desc")= chr  "distance argument r" "theoretical Poisson %s" "Chiu-Stoyan estimate of %s"
##  - attr(*, "units")=List of 3
##   ..$ singular  : chr "unit"
##   ..$ plural    : chr "units"
##   ..$ multiplier: num 1
##   ..- attr(*, "class")= chr "units"
##  - attr(*, "fname")= chr "F"
##  - attr(*, "dotnames")= chr  "km" "rs" "cs" "theo"
```

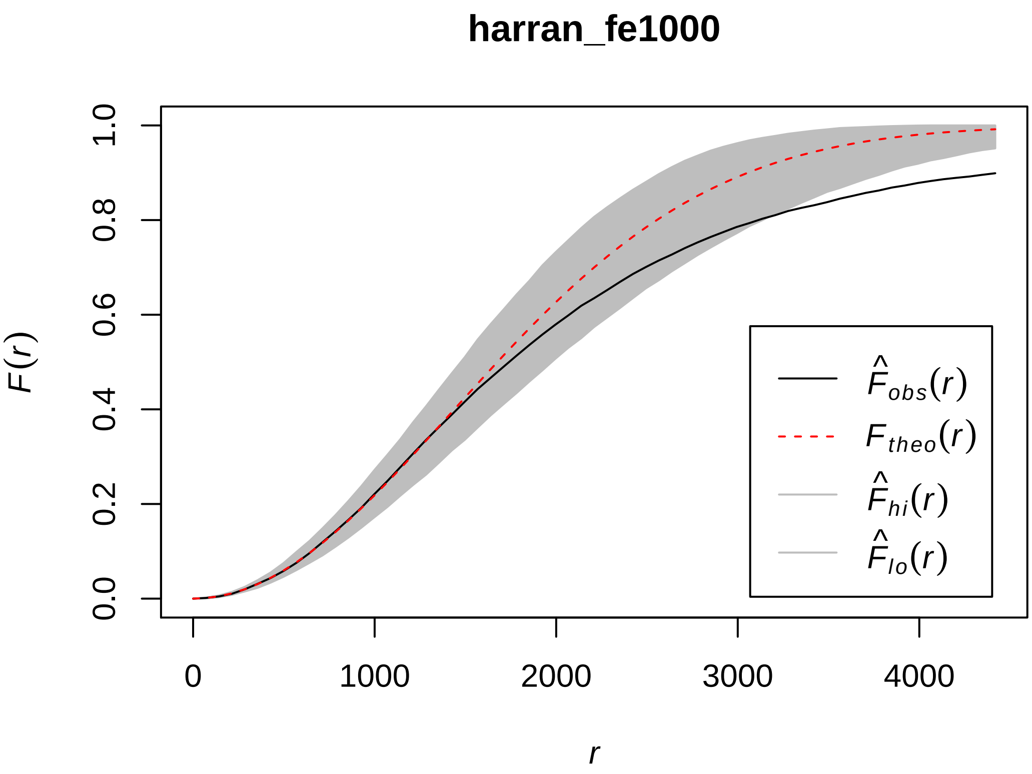
```
plot(harran_f)
```



```
harran_fe1000 <- envelope(harran_ppp2, fun="Fest", nsim=1000)
```

```
## Generating 1000 simulations of CSR  ...
## 1, 2, 3, ......10.........20.........30.........40.........50.........60...
## ......70.........80.........90.........100.........110.........120......
## ...130.........140.........150.........160.........170.........180.........
## 190.........200.........210.........220.........230.........240.........250..
## .......260.........270.........280.........290.........300.........310.....
## ....320.........330.........340.........350.........360.........370........
## .380.........390.........400.........410.........420.........430.........440.
## ........450.........460.........470.........480.........490.........500....
## .....510.........520.........530.........540.........550.........560.......
## ..570.........580.........590.........600.........610.........620.........630
## .........640.........650.........660.........670.........680.........690...
## ......700.........710.........720.........730.........740.........750......
## ...760.........770.........780.........790.........800.........810.........
## 820.........830.........840.........850.........860.........870.........880..
## .......890.........900.........910.........920.........930.........940.....
## ....950.........960.........970.........980.........990......... 1000.
## 
## Done.
```

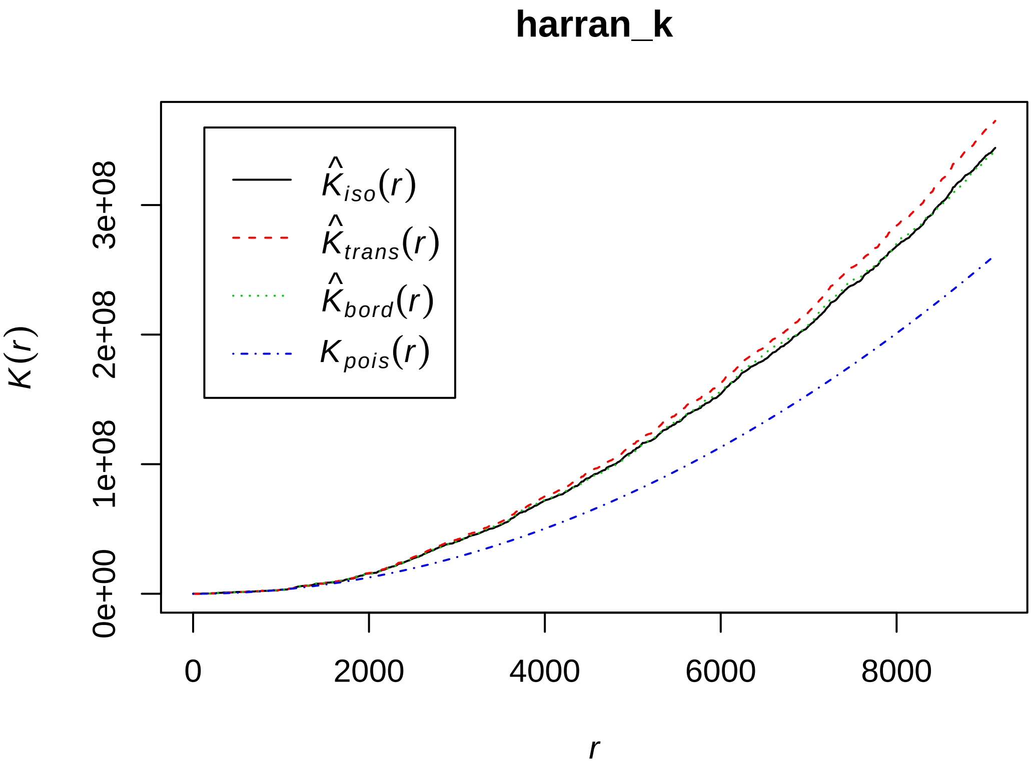
```
plot(harran_fe1000)
```



```r
# K-Function
harran_k <- Kest(harran_ppp2)
str(harran_k)
```

```
## Classes 'fv' and 'data.frame':   513 obs. of  5 variables:
##  $ r     : num  0 17.8 35.6 53.4 71.3 ...
##  $ theo  : num  0 997 3989 8975 15956 ...
##  $ border: num  0 0 0 0 0 ...
##  $ trans : num  0 0 0 0 0 ...
##  $ iso   : num  0 0 0 0 0 ...
##  - attr(*, "argu")= chr "r"
##  - attr(*, "valu")= chr "iso"
##  - attr(*, "ylab")= language K(r)
##  - attr(*, "yexp")= language K(r)
##  - attr(*, "fmla")= chr ".~r"
##  - attr(*, "alim")= num  0 9122
##  - attr(*, "labl")= chr  "r" "%s[pois](r)" "hat(%s)[bord](r)" "hat(%s)[trans](r)" ...
##  - attr(*, "desc")= chr  "distance argument r" "theoretical Poisson %s" "border-corrected estimate o
##  - attr(*, "units")=List of 3
##   ..$ singular  : chr "unit"
##   ..$ plural    : chr "units"
##   ..$ multiplier: num 1
##   ..- attr(*, "class")= chr "units"
##  - attr(*, "fname")= chr "K"
##  - attr(*, "dotnames")= chr  "iso" "trans" "border" "theo"
```

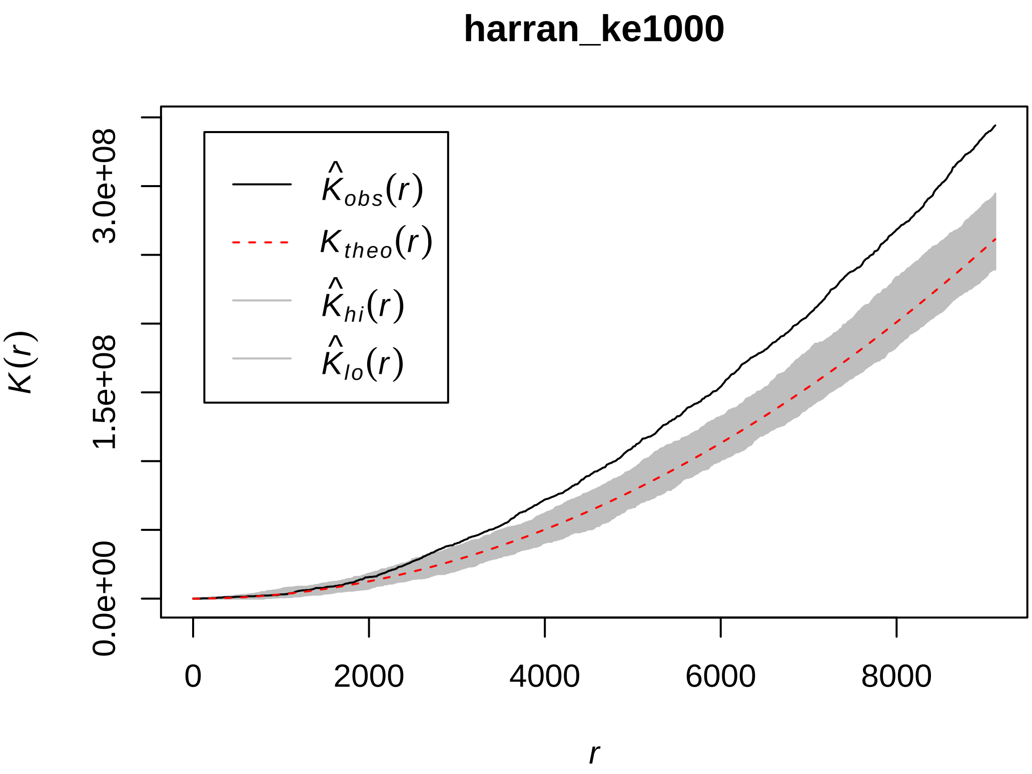
```r
plot(harran_k)
```



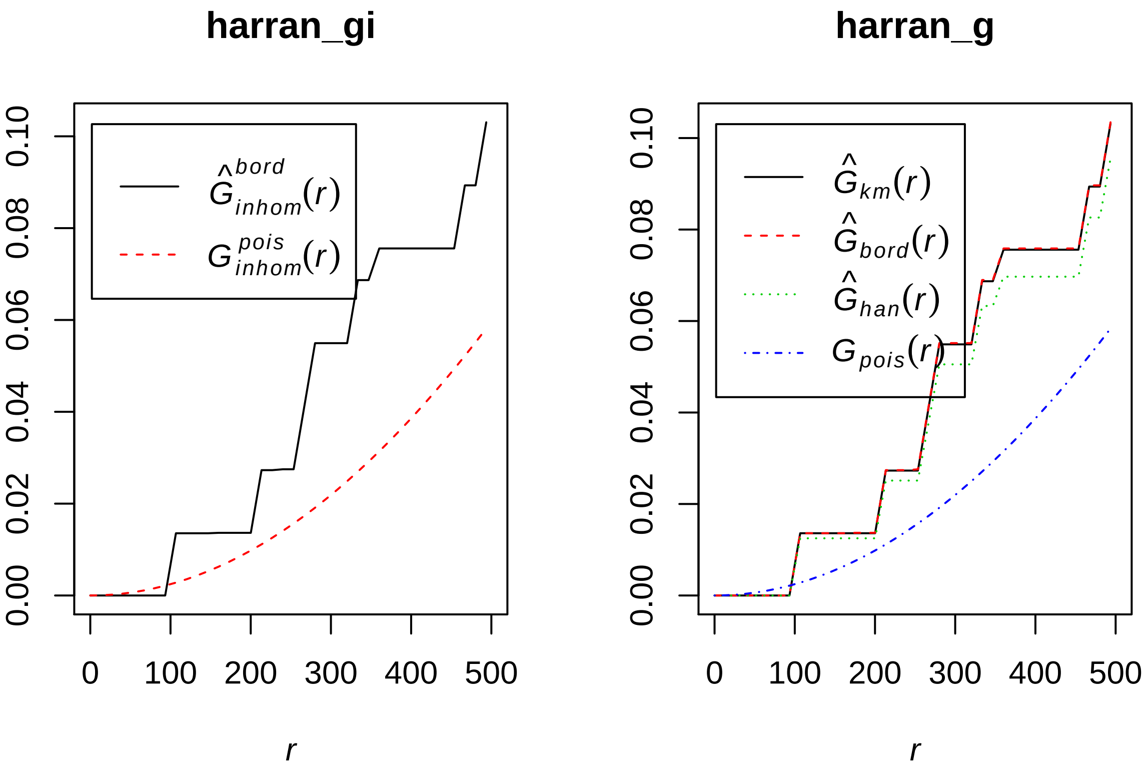
```
harran_ke1000 <- envelope(harran_ppp2, fun="Kest", nsim=1000)
```

```
## Generating 1000 simulations of CSR  ...
## 1, 2, 3, ......10.........20.........30.........40.........50.........60...
## ......70.........80.........90.........100.........110.........120......
## ...130.........140.........150.........160.........170.........180.........
## 190.........200.........210.........220.........230.........240.........250..
## .......260.........270.........280.........290.........300.........310.....
## ....320.........330.........340.........350.........360.........370........
## .380.........390.........400.........410.........420.........430.........440.
## ........450.........460.........470.........480.........490.........500....
## .....510.........520.........530.........540.........550.........560.......
## ..570.........580.........590.........600.........610.........620.........630
## .........640.........650.........660.........670.........680.........690...
## ......700.........710.........720.........730.........740.........750......
## ...760.........770.........780.........790.........800.........810.........
## 820.........830.........840.........850.........860.........870.........880..
## .......890.........900.........910.........920.........930.........940.....
## ....950.........960.........970.........980.........990......... 1000.
##
## Done.
```

```
plot(harran_ke1000)
```



```
# Inhomogeneous G/F/K:
harran_gi <- Ginhom(harran_ppp2, lambda=predict(harran_rhohat))
par(mfrow=c(1,2))
plot(harran_gi, xlim=c(0,500))
plot(harran_g, xlim=c(0,500))
```



```
#Finhom()
#Kinhom()
```

Hijmans, Robert J. 2016. *Raster: Geographic Data Analysis and Modeling*. https://CRAN.R-project.org/package=raster.

SCHAUER, ROLAND, and HANS FAILLARD. 1968. "Zur Wirkungsspezifität Der Neuraminidase. Das Verhalten Isomerer No-Diacetyl-Neuraminsäureglykoside Im Submaxillarismucin von Pferd Und Rind Bei Einwirkung Bakterieller Neuraminidase." *Hoppe-Seyler' S Zeitschrift Für Physiologische Chemie* 349 (2): 961–68.