HarranPlain

Nicole Grunert

20 Juli 2017

## Point Pattern

harran=read.table("../data/Sites\_HarranPlain.csv",  
 sep = ",",  
 header = TRUE) # when knitting: "../data/Sites\_HarranPlain.csv"!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!  
str(harran)

## 'data.frame': 344 obs. of 5 variables:  
## $ X.1 : int 1 2 3 4 5 6 7 8 9 10 ...  
## $ Name : Factor w/ 166 levels "Ãmertepe","ÃncÃ¼l (FALSCH)",..: 24 10 10 67 67 67 67 78 78 78 ...  
## $ X : num 38.8 38.9 38.9 38.9 38.9 ...  
## $ Y : num 37.6 37.7 37.7 37.2 37.2 ...  
## $ Mentioned\_Epoch: Factor w/ 179 levels "","-","Aceramic Neolithic ",..: 175 150 139 162 108 151 161 134 125 86 ...

## spatstat

library(sp)

## Warning: package 'sp' was built under R version 3.3.3

coordinates(harran) <- ~X+Y  
proj4string(harran) <- CRS("+init=epsg:4326")  
  
harran <- spTransform(harran, CRSobj = CRS("+init=epsg:32637"))  
str(harran) # for checking

## Formal class 'SpatialPointsDataFrame' [package "sp"] with 5 slots  
## ..@ data :'data.frame': 344 obs. of 3 variables:  
## .. ..$ X.1 : int [1:344] 1 2 3 4 5 6 7 8 9 10 ...  
## .. ..$ Name : Factor w/ 166 levels "Ãmertepe","ÃncÃ¼l (FALSCH)",..: 24 10 10 67 67 67 67 78 78 78 ...  
## .. ..$ Mentioned\_Epoch: Factor w/ 179 levels "","-","Aceramic Neolithic ",..: 175 150 139 162 108 151 161 134 125 86 ...  
## ..@ coords.nrs : num(0)   
## ..@ coords : num [1:344, 1:2] 479412 486771 486771 493122 493122 ...  
## .. ..- attr(\*, "dimnames")=List of 2  
## .. .. ..$ : chr [1:344] "1" "2" "3" "4" ...  
## .. .. ..$ : chr [1:2] "X" "Y"  
## ..@ bbox : num [1:2, 1:2] 477942 4062337 514430 4290885  
## .. ..- attr(\*, "dimnames")=List of 2  
## .. .. ..$ : chr [1:2] "X" "Y"  
## .. .. ..$ : chr [1:2] "min" "max"  
## ..@ proj4string:Formal class 'CRS' [package "sp"] with 1 slot  
## .. .. ..@ projargs: chr "+init=epsg:32637 +proj=utm +zone=37 +datum=WGS84 +units=m +no\_defs +ellps=WGS84 +towgs84=0,0,0"

library(spatstat)

## Warning: package 'spatstat' was built under R version 3.3.3

## Loading required package: nlme

## Warning: package 'nlme' was built under R version 3.3.3

## Loading required package: rpart

## Warning: package 'rpart' was built under R version 3.3.3

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

##   
## Note: R version 3.3.1 (2016-06-21) is more than 9 months old; we strongly recommend upgrading to the latest version

str(harran@coords) # structure

## num [1:344, 1:2] 479412 486771 486771 493122 493122 ...  
## - attr(\*, "dimnames")=List of 2  
## ..$ : chr [1:344] "1" "2" "3" "4" ...  
## ..$ : chr [1:2] "X" "Y"

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

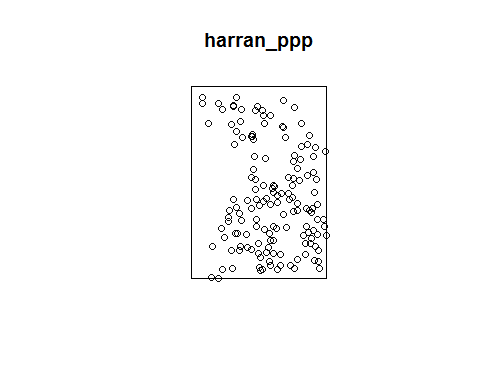
## Warning: 65 points were rejected as lying outside the specified window

## Warning: data contain duplicated points

harran\_ppp=unique.ppp(harran\_ppp) # shows number of duplicated points and deletes them/ harran\_ppp= has to be done to define harran\_ppp new  
  
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 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\_ppp)



library(mapview)

## Warning: package 'mapview' was built under R version 3.3.3

## Loading required package: leaflet

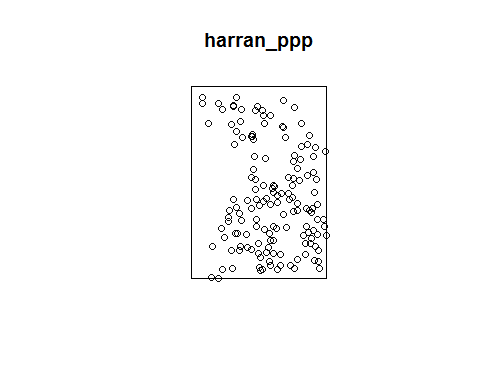
## Warning: package 'leaflet' was built under R version 3.3.3

mapview(harran)

## PhantomJS not found. You can install it with webshot::install\_phantomjs(). If it is installed, please make sure the phantomjs executable can be found via the PATH variable.

## Challenge: delete duplicated points

harran\_ppp=unique.ppp(harran\_ppp) # shows number of duplicated points and deletes them/ harran\_ppp= has to be done to define harran\_ppp new  
  
  
# or:  
#anyDuplicated(harran\_ppp)  
#harran <- unique(harran\_ppp)  
#harran\_ppp <- harran\_ppp[!duplicated(harran\_ppp)]  
  
plot(harran\_ppp)

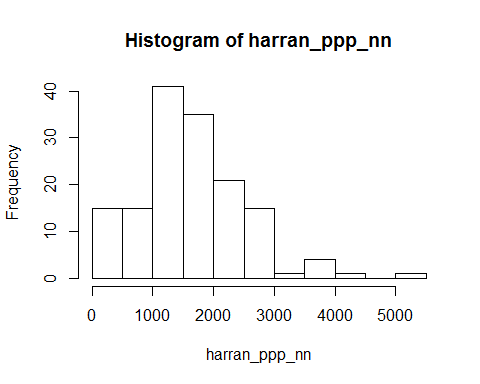


# Nearest neighbour distance

harran\_ppp\_nn <- nndist(harran\_ppp)  
str(harran\_ppp\_nn) # shows distance within the structure(str)

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

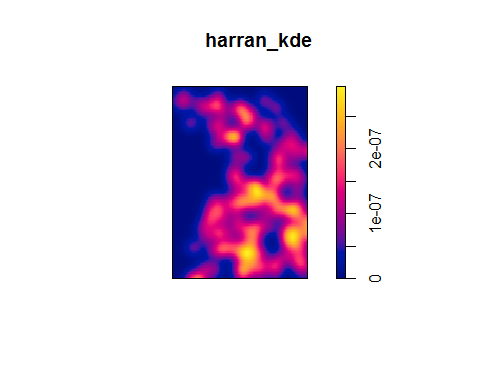
hist(harran\_ppp\_nn) # plots the nearest neighbour



#barplot(sort(harran\_ppp\_nn))

# challenge: create a kernel density estimation

harran\_kde <- density.ppp(harran\_ppp, sigma = mean(harran\_ppp\_nn))# see: likelihood cross validation bandwidth selection for kernel density (help)  
plot(harran\_kde)



# raster

library(raster)

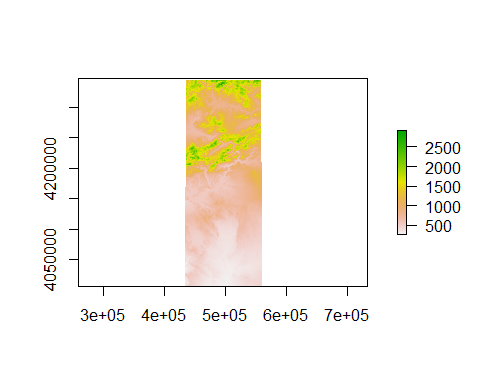
## Warning: package 'raster' was built under R version 3.3.3

##   
## Attaching package: 'raster'

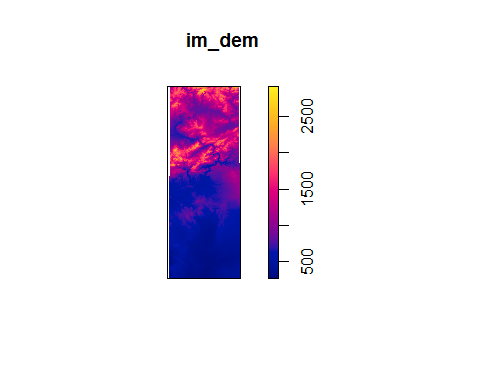
## The following objects are masked from 'package:spatstat':  
##   
## area, rotate, shift

## The following object is masked from 'package:nlme':  
##   
## getData

dem <- raster("../data/dem.tif") # see above for problems when knitting  
  
# or: library(rgdal)  
#dem <- readGDAL("data/dem.tif")  
plot(dem)

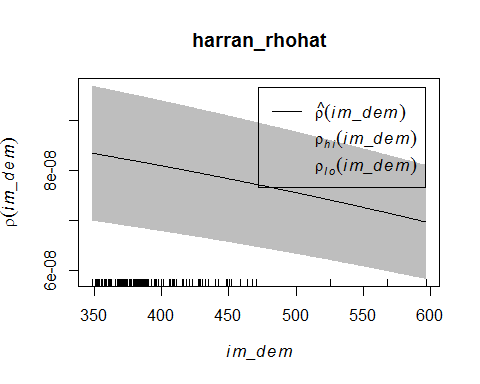


im\_dem <- as.im(as.image.SpatialGridDataFrame(as(dem, "SpatialGridDataFrame"))) #creates image  
plot(im\_dem)

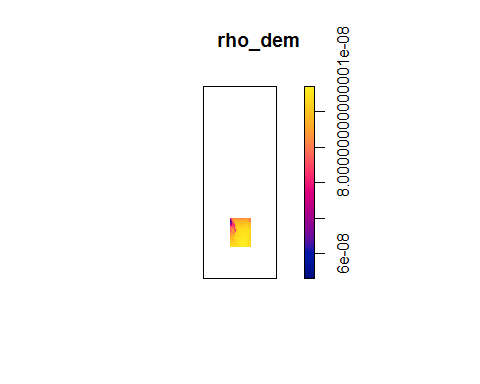


# challenge: use rhohat and create a plot

#?rhohat # smoothing estimate: changes the raster  
harran\_rhohat <- rhohat(harran\_ppp, im\_dem, bw = 200)  
 # <- rhohat(harran\_ppp, im\_dem, bw=200) /gives a more distinct picture  
  
plot(harran\_rhohat) #x=elevation y=relative intensity of points -> relation of elevation to pointdensity, bandwidth=default -> default=sigma in the structure of the object(str(harran\_rhohat))



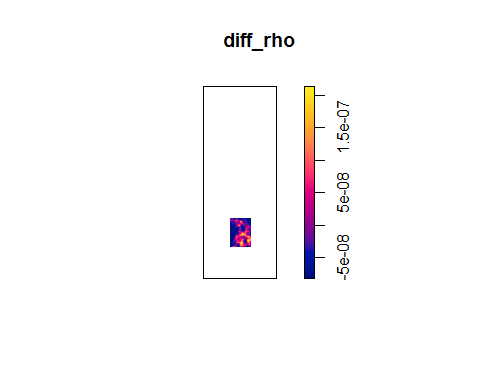
rho\_dem <- predict(harran\_rhohat)  
plot(rho\_dem)



diff\_rho <- harran\_kde-rho\_dem

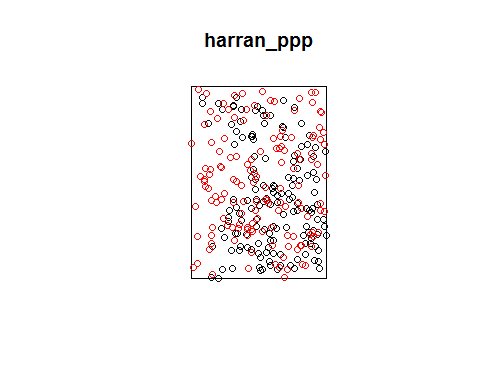
## Warning: the images 'e1' and 'e2' were not compatible

plot(diff\_rho)



# challenge: test poisson, create random points with rpoispp function that have the same intensity like our points

set.seed(123)  
harran\_rpoispp2 <- rpoispp(lambda = harran\_ppp$n/area.owin(harran\_ppp$window), win=harran\_ppp$window)  
set.seed(123)  
harran\_rpoispp3 <- rpoispp(intensity(harran\_ppp), win=Window(harran\_ppp))  
set.seed(123)  
harran\_rpoispp4 <- rpoispp(ex = harran\_ppp)  
  
plot(harran\_ppp)  
points(harran\_rpoispp2, col="green")  
points(harran\_rpoispp3, col="blue")  
points(harran\_rpoispp4, col="red")



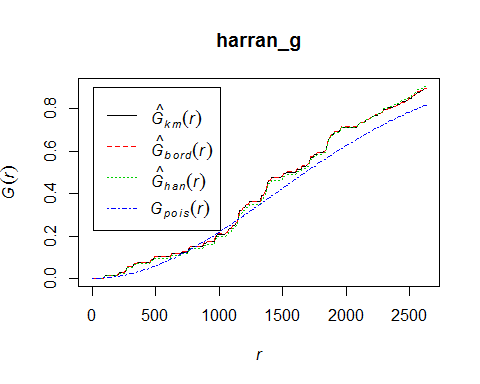
# first block is all the same, different ways to get the same result

# Second order effects

harran\_g <- Gest(harran\_ppp)  
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" "border corrected estimate of %s" ...  
## - 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"

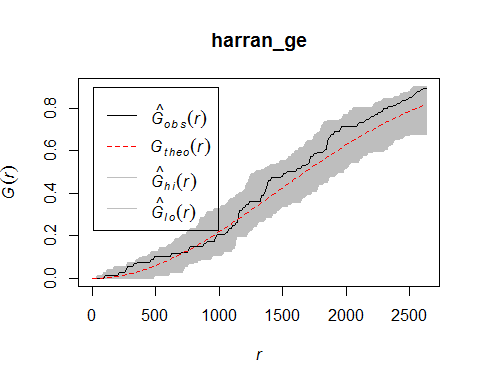
plot(harran\_g) # x=closest neighbours expected (blue), the rest shows higher than expected clusters y= distance



harran\_ge <- envelope(harran\_ppp, fun = "Gest") # calculates g function for random points

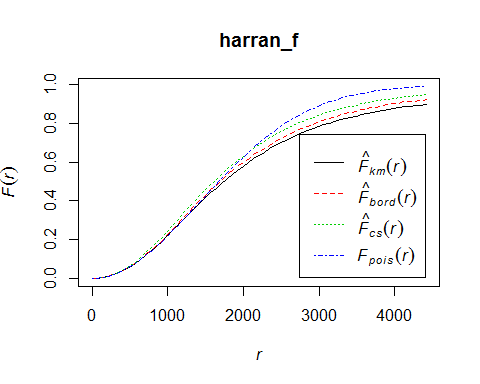
## Generating 99 simulations of CSR ...  
## 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 36, 37, 38,  
## 39, 40, 41, 42, 43, 44, 45, 46, 47, 48, 49, 50, 51, 52, 53, 54, 55, 56, 57, 58, 59, 60, 61, 62, 63, 64, 65, 66, 67, 68, 69, 70, 71, 72, 73, 74, 75, 76,  
## 77, 78, 79, 80, 81, 82, 83, 84, 85, 86, 87, 88, 89, 90, 91, 92, 93, 94, 95, 96, 97, 98, 99.  
##   
## Done.

plot(harran\_ge) # grey shadow\_ monte Carlo Simulation



# challenge: do F and K Function

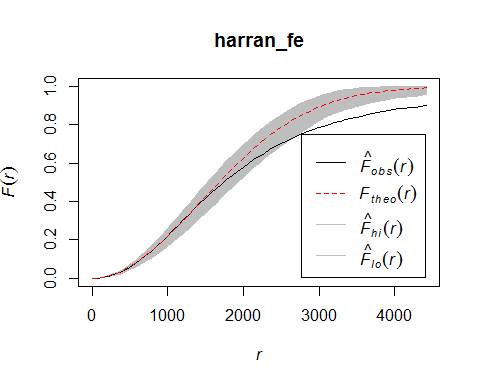
#F-function:  
  
harran\_f <- Fest(harran\_ppp)  
plot(harran\_f)



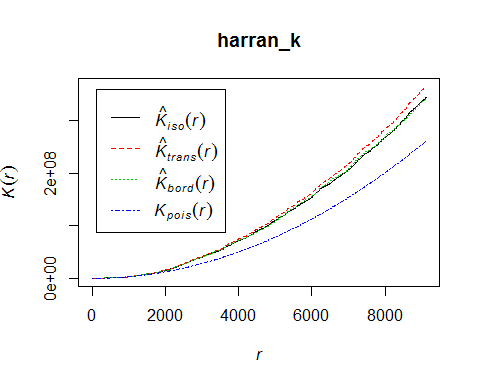
harran\_fe <- envelope(harran\_ppp, fun = "Fest") # calculates f function for random points

## 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, 29, 30, 31, 32, 33, 34, 35, 36, 37, 38,  
## 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, 64, 65, 66, 67, 68, 69, 70, 71, 72, 73, 74, 75, 76,  
## 77, 78, 79, 80, 81, 82, 83, 84, 85, 86, 87, 88, 89, 90, 91, 92, 93, 94, 95, 96, 97, 98, 99.  
##   
## Done.

plot(harran\_fe)



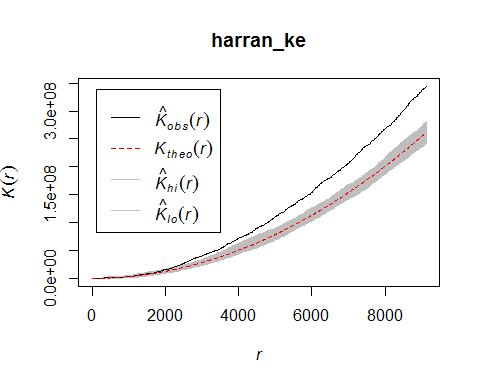
## red: expected, black deviates -> expect that the empty spaces are smaller than expected = clustered  
  
  
#K-function  
  
harran\_k <- Kest(harran\_ppp)  
plot(harran\_k)



harran\_ke <- envelope(harran\_ppp, fun = "Kest")

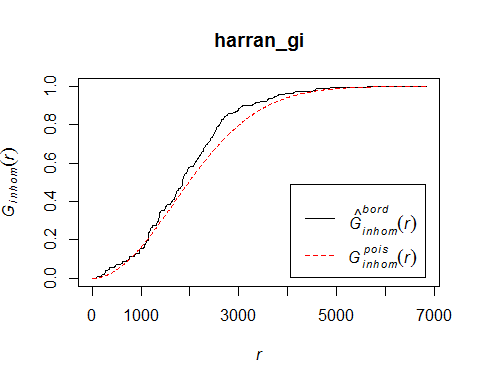
## Generating 99 simulations of CSR ...  
## 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 36, 37, 38,  
## 39, 40, 41, 42, 43, 44, 45, 46, 47, 48, 49, 50, 51, 52, 53, 54, 55, 56, 57, 58, 59, 60, 61, 62, 63, 64, 65, 66, 67, 68, 69, 70, 71, 72, 73, 74, 75, 76,  
## 77, 78, 79, 80, 81, 82, 83, 84, 85, 86, 87, 88, 89, 90, 91, 92, 93, 94, 95, 96, 97, 98, 99.  
##   
## Done.

plot(harran\_ke)

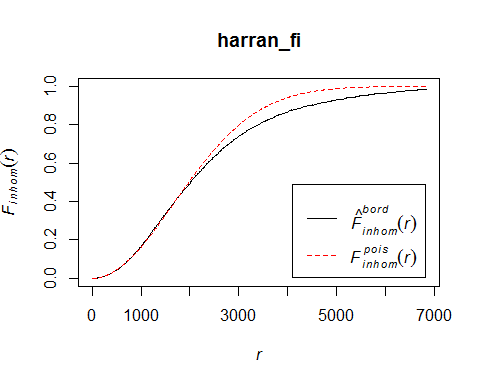


# Inhomogeneous Poissonfunction G/F/K

harran\_gi <- Ginhom(harran\_ppp, lambda = predict(harran\_rhohat)) # harran\_rhohat needs an bandwidth of 200  
plot(harran\_gi)



harran\_fi <- Finhom(harran\_ppp, lambda = predict(harran\_rhohat))  
plot(harran\_fi)



#par(mfrow = c(1,2))  
#plot(harran\_gi, xlim = c(0,6000))  
#plot(harran\_g, xlim = c(0,6000)) Gegenüberstellung

Note that the echo = FALSE parameter is added to the code chunk to prevent printing of the R code that generated the plot.