Supplementary Rmarkdown for Davis et al. 'The integration of lidar and legacy datasets provides improved explanations for the spatial patterning of shell rings in the American Southeast'

Robert J. DiNapoli

3/1/2020

INTRODUCTION

This document provides a reproducible spatial analysis for shell ring features in Beaufort County, S.C.

LOAD PACKAGES

```
library(spatstat)
library(maptools)
library(raster)
library(rgdal)
library(rgeos) library(sp)
library(MuMIn)
library(here)
```

LOAD WORKSPACE

```
setwd(here())
```

LOAD DATA

```
rings_shp <- readOGR(".", "rings_WGS84") #load all shell rings data set
c_rings_shp <- readOGR(".", "confirmed_rings_WGS84") #load confirmed shell
rings data set
l_rings_shp <-readOGR(".", "LiDAR_rings_WGS") #read in shapefile b_win_shp
<- readOGR(".", "Beaufort_Boundary_WGS84_17N") #load boundary window
elev <- mask(raster("DEM_fill_WGS_3m_x3.tif"),b_win_shp) #clip raster to be
within survey area
elev[elev < 0] <- 0 #change negative values to 0
water_shp <- readOGR(".", "water_clip_WGS84") #load water dataset
water_shp_clip <- gIntersection(water_shp, b_win_shp, byid=T) #clip water to
boundary window

soils <- raster("soils_perm_rank.tif") #load soils dataset
```

CONVERT TO SPATSTAT FORMAT

```
b_win <- as.owin(b_win_shp) #convert to window object
rings_pp <- ppp(rings_shp$POINT_X, rings_shp$POINT_Y, window=b_win) #convert
to pp object c_rings_pp <- ppp(c_rings_shp$POINT_X, c_rings_shp$POINT_Y,
window=b_win)
#convert to pp object
l_rings_pp <- ppp(l_rings_shp$POINT_X, l_rings_shp$POINT_Y, window=b_win) elev
<- as.im(elev) #convert to pixel image
water <- as.owin(water_shp_clip) #convert to window object
water_dist <- distfun(water) #convert to Euclidean distance
function water_dist <- as.im(water_dist, W=b_win) #convert to pixel
image soils <- as.im(soils) #convert to pixel image soils <-
as.im(soils, W=b_win) #clip to window
soils_quad <- as.tess(soils) #convert to tesselation
```

EXPLORATORY DATA ANALYSIS

Kernel density estimates

Compute kernel density estimates for shell rings using likelihood cross-validation to select a smoothing bandwidth.

```
KDE_c_rings <- density(c_rings_pp, bw.ppl) #confirmed rings
KDE_l_rings <- density(l_rings_pp, bw.ppl) #lidar rings
KDE_rings <- density(rings_pp, bw.ppl) #all features
```

Nearest neighbors

Compute nearest neighbor distances between points.

```
c_rings_nn <- nndist(c_rings_pp) #confirmed rings
l_rings_nn <- nndist(l_rings_pp) #lidar rings rings_nn
<- nndist(rings_pp) #all features
```

Pair correlation function

Compute the pair-correlation function with 999 Monte Carlo simulations of CSR (p=0.002). Because point pattern is likely not isotropic and interaction occurs at large distances, the translation edge correction is used (Baddeley et al. 2015:218,528). This method also implements a distance divisor following Baddeley et al. (2015:229).

```
rings_pcf <- envelope(rings_pp, fun=pcf, divisor="d",
fit.n=T,correction="translation", nsim=999)
c_rings_pcf <- envelope(c_rings_pp, fun=pcf, divisor="d",
fit.n=T,correction="translation", nsim=999) l_rings_pcf
<- envelope(l_rings_pp, fun=pcf, divisor="d",

fit.n=T,correction="translation", nsim=999)
```

Dependence of point pattern on spatial covariates

This code chunk creates Figure 5

```
par(mfrow=c(2,2)) par(mai=c(0,0,0.2,0))
plot(KDE_rings, main="Shell rings KDE", col=gray.colors(5, start = 0.3, end =
0.9, gamma = 2.2, rev = FALSE), riblab="Density") plot(c_rings_pp,
pch=15, cex=1, col='red', add=T) plot(l_rings_pp, pch=16, cex=1,
col='blue', add=T) par(mai=c(0,0,0.2,0.2))
plot(elev, main="Elevation", col=gray.colors(5, start = 0.3, end = 0.9, gamma
= 2.2, rev = FALSE), riblab="MASL") plot(b_win,
add=T) plot(c_rings_pp, pch=15, cex=1, col='red',
add=T) plot(l_rings_pp, pch=16, cex=1, col='blue',
add=T) par(mai=c(0,0,0.2,0))
plot(water_dist, main="Distance to water", col=gray.colors(8, start = 0.3,
end = 0.9, gamma = 2.2, rev = FALSE), riblab="Meters") plot(b_win, add=T)
plot(c_rings_pp, pch=15, cex=1, col='red', add=T) plot(l_rings_pp, pch=16,
cex=1, col='blue', add=T) par(mai=c(0,0,0.2,0.2))
plot(soils_quad, main="Soil Permeability", col=gray.colors(4, start = 0.3,
end = 0.9, gamma = 2.2, rev = FALSE), riblab="Class") plot(b_win, add=T)
plot(c_rings_pp, pch=15, cex=1, col='red', add=T) plot(l_rings_pp, pch=16,
cex=1, col='blue', add=T) par(mfrow=c(1,1)) dev.off()
```

Use nonparametric regression (rhohat) to examine the intensity relative to elevation and distance from water (See Baddeley et al. 2015:180-183 for details). As soils are factor valued, we simply calculated the count of shell rings within each soil permeability class.

```
c_elev_rh <- rhohat(c_rings_pp, elev)#intensity as a function of elevation,
with 95% confidence bands
c_water_rh <- rhohat(c_rings_pp, water_dist)#intensity as a function of
water, with 95% confidence bands
c_soils_quad_count <- quadratcount(c_rings_pp, tess=soils_quad)#compute the
number of shell rings within each soil permeability ranking l_elev_rh <-
rhohat(l_rings_pp, elev) l_water_rh <- rhohat(l_rings_pp, water_dist)
l_soils_quad_count <- quadratcount(l_rings_pp, tess=soils_quad)
elev_rh <- rhohat(rings_pp, elev)  water_rh <- rhohat(rings_pp,
water_dist)
soils_quad_count <- quadratcount(rings_pp, tess=soils_quad)
```

This code chunk creates Figure 6.

```
par(mfrow=c(5,3))
hist(c_rings_nn, xlab="Nearest Neighbors (meters)", main="Confirmed Rings",
breaks=8, col="grey", xlim=c(0,20000), ylim=c(0,22))
mtext(side=3, line=-0.25, at=-0.07, adj=0, cex=0.9, "(a) Histogram of nearest
neighbors")
hist(l_rings_nn, xlab="Nearest Neighbors (meters)", main="LiDAR Rings",
breaks=8, col="grey", xlim=c(0,20000), ylim=c(0,22))
hist(rings_nn, xlab="Nearest Neighbors (meters)", main="All", breaks=8,
col="grey", xlim=c(0,20000), ylim=c(0,22))
plot(c_rings_pcf, main="",ylim=c(0,15),xlim=c(0,8000), legend=F, xlab=("r
(meters)"))
mtext(side=3, line=1, at=-0.07, adj=0, cex=0.9, "(b) Pair correlation
function")
plot(l_rings_pcf, main="",ylim=c(0,6),xlim=c(0,8000), legend=F, xlab=("r
(meters)"))
plot(rings_pcf, main="",ylim=c(0,6),xlim=c(0,8000), legend=F, xlab=("r
(meters)"))
plot(c_elev_rh, legend=F, xlab="Elevation (masl)", main="")
mtext(side=3, line=1, at=-0.07, adj=0, cex=0.9, "(c) Rings as a function
of elevation") plot(l_elev_rh, legend=F, xlab="Elevation (masl)", main='')
plot(elev_rh, legend=F, xlab="Elevation (masl)", main='') plot(c_water_rh,
legend=F, main="", xlab="Distance to water (m)", xlim=c(0,4000))
mtext(side=3, line=1, at=-0.07, adj=0, cex=0.9, "(d) Rings as a function of
water")
plot(l_water_rh, legend=F, main="", xlab="Distance to water (m)",
xlim=c(0,4000))
plot(water_rh, legend=F, main="", xlab="Distance to water (m)",
xlim=c(0,4000)) barplot(c_soils_quad_count, ylab="Count",
xlab="Permeability ranking") mtext(side=3, line=1, at=-0.07, adj=0,
cex=0.9, "(e) Rings as a function of soil class")
barplot(l_soils_quad_count, ylab="Count", xlab="Permeability ranking")
barplot(soils_quad_count, ylab="Count", xlab="Permeability ranking")
par(mfrow=c(1,1)) dev.off()
```

POINT PROCESS MODELING

Here we fit a series of point process models to the total shell rings point pattern to assess which variables and parameters best account for the patterns. We begin with a null model (ppm0), in this case that the underlying process is complete spatial randomness (CSR), aka a homogeneous Poisson process. We then fit a series of models that contain different combinations of the variables of interest (ppm1 through ppm7). The default edge correction for ppm is 'border', but here we use 'translation' given interaction distances and anisotropy.

```
ppm0 <- ppm(rings_pp, ~1, correction="translation") ppm1 <-
ppm(rings_pp, ~soils_quad, correction="translation" ) ppm2 <-
ppm(rings_pp, ~elev, correction="translation") ppm3 <- ppm(rings_pp,
~water_dist, correction="translation") ppm4 <- ppm(rings_pp,
~soils_quad+elev, correction="translation") ppm5 <- ppm(rings_pp,
~soils_quad+water_dist, correction="translation") ppm6 <- ppm(rings_pp,
~elev+water_dist, correction="translation")
ppm7 <- ppm(rings_pp, ~soils_quad+elev+water_dist, correction="translation")
```

We then use AICc and BIC to compare each model.

```
MS1_AIC <- model.sel(ppm0, ppm1, ppm2, ppm3, ppm4, ppm5, ppm6, ppm7,
rank=AICc)
MS1_BIC <- model.sel(ppm0, ppm1, ppm2, ppm3, ppm4, ppm5, ppm6, ppm7,
rank=BIC)
```

This code chunk yields the results in Table 3

```
MS1_AIC
```

```
## Model selection table
##      sls_qud     trend df   logLik   AICc delta weight
## ppm2                 e  2 -946.931 1898.1  0.00  0.635
## ppm6             e+w_d  3 -946.854 1900.2  2.10  0.222
## ppm4       +     s_q+e  5 -945.185 1901.7  3.57  0.107
## ppm7       + s_q+e+w_d  6 -945.060 1904.0  5.88  0.034
## ppm1       +       s_q  4 -950.700 1910.3 12.15  0.001
## ppm5       +   s_q+w_d  5 -950.305 1911.9 13.81  0.001
## ppm3               w_d  2 -959.384 1923.0 24.91  0.000
## ppm0                    1 -961.858 1925.8 27.69  0.000 ##
Abbreviations:
## trend:  = '~1', e = '~elev', e+w_d = '~elev+water_dist',
##        s_q = '~soils_quad', s_q+e = '~soils_quad+elev',
##        s_q+e+w_d = '~soils_quad+elev+water_dist',
##        s_q+w_d = '~soils_quad+water_dist', w_d = '~water_dist'
## Models ranked by AICc(x)
```

```
MS1_BIC
```

```
## Model selection table
##      sls_qud     trend df   logLik    BIC delta weight
## ppm2                 e  2 -946.931 1901.8  0.00  0.856
## ppm6             e+w_d  3 -946.854 1905.6  3.80  0.128
## ppm4       +     s_q+e  5 -945.185 1910.1  8.36  0.013
## ppm7       + s_q+e+w_d  6 -945.060 1913.8 12.06  0.002
## ppm1       +       s_q  4 -950.700 1917.2 15.44  0.000
## ppm5       +   s_q+w_d  5 -950.305 1920.4 18.60  0.000
## ppm3               w_d  2 -959.384 1926.7 24.91  0.000
## ppm0                    1 -961.858 1927.7 25.90  0.000 ##
Abbreviations:
## trend:  = '~1', e = '~elev', e+w_d = '~elev+water_dist',
```

```
##         s_q = '~soils_quad', s_q+e = '~soils_quad+elev',
##         s_q+e+w_d = '~soils_quad+elev+water_dist',
##         s_q+w_d = '~soils_quad+water_dist', w_d = '~water_dist'
## Models ranked by BIC(x)
```

Both model selection crieria suggest that the inhomogeneous model ppm2 that models shell rings as a log-linear function of elevation fits best. To assess the fit between this model and the data, in particular to check for a second-order interaction component, we compare the empirical pattern to expectations from the model by generating 99 simulated realizations of the residual K- and G- functions.

```
K_sim1 <- envelope(ppm2, Kres, nsim=99, fix.n=T, correction="translation")
G_sim1 <- envelope(ppm2, Gres, nsim=99, fix.n=T, correction="best")
```

This code chunk creates Figure 7

```
par(mfrow=c(1,2)) plot(K_sim1, xlim=c(0,4000),
main="", legend=F) plot(G_sim1, xlim=c(0,4000),
main="",legend=F) par(mfrow=c(1,1))
```

Both tests indicate that the empirical point pattern is more clustered than is accounted for by model ppm2. We then add an interaction term to the model to account for interpoint clustering. Here we use a Gibbs Area Interaction process with irregular parameter r=2000 based on nearest neighbor distances and the pair correlation function. The first model (ppm8) only includes the clustering parameter to test the hypothesis that the point pattern is simply explained by inter-point clustering and not elevation. The second model (ppm9) includes Area Interaction and elevation.

```
ppm8 <- ppm(rings_pp, ~ 1, AreaInter(2000))
ppm9 <- ppm(rings_pp, ~ elev, AreaInter(2000))
```

We then use AICc and BIC to select the best-fitting model.

```
MS2_AIC <- model.sel(ppm2, ppm8, ppm9, rank=AICc)

## Warning: log likelihood is not available for non-Poisson model; log
## pseudolikelihood returned

MS2_BIC <- model.sel(ppm2, ppm8, ppm9, rank=BIC)
```

This code chunk yields the results in Table 4. Both model selection criteria indicate that the model with Area Interaction and elevation (ppm9) fits best.

```
MS2_AIC

## Model selection table
##      trend correction   interaction df   logLik   AICc  delta weight
## ppm9  elev            AreIntr(2000)  3 -676.613 1359.7   0.00  0.969
## ppm8                  AreIntr(2000)  2 -681.170 1366.6   6.86  0.031
## ppm2  elev translatin                2 -946.931 1898.1 538.38  0.000 ##
Abbreviations:
```

```
## trend:  = '~1', elev = '~elev'
## correction: translatin = 'translation'
## interaction: AreIntr(2000) = 'AreaInter(2000)' ##
Models ranked by AICc(x)

MS2_BIC

## Model selection table
##      trend correction   interaction df   logLik    BIC  delta weight
## ppm9  elev            AreIntr(2000)  3 -676.613 1365.1   0.00   0.93
## ppm8                  AreIntr(2000)  2 -681.170 1370.2   5.16   0.07
## ppm2  elev translatin                2 -946.931 1901.8 536.68   0.00
## Abbreviations:
## trend:  = '~1', elev = '~elev'
## correction: translatin = 'translation'
## interaction: AreIntr(2000) = 'AreaInter(2000)'
## Models ranked by BIC(x)
```

This code chunk yields the results in Table 5 - the parameter estimates for ppm9 and standard errors with 95% confidence intervals.

```
coef(summary(ppm9))

##                 Estimate       S.E.      CI95.lo      CI95.hi Ztest
## (Intercept) -18.56898494 0.32812007 -19.21208845 -17.92588143   ***
## elev          0.04497521 0.01274398   0.01999747   0.06995294   *** ##
Interaction   2.03385337 0.54333298   0.96894029   3.09876645   *** ##
Zval
## (Intercept) -56.592043
## elev          3.529134
## Interaction   3.743291
```

To assess the fit between the interaction component of this model and the data we generate 99 simulated realizations of the residual K- and G- functions. We also compute the lurkingvariable and partial-residual plots to assess any deviations between the empirical intensity of shell rings and their fitted intensity as a function of elevation. The lurking variable plot shows the relationship between the residuals and elevation, and the partial residual plot shows the relationship between the estimated intensity as a function of elevation and a smoothed estimate of its empirical effect.

```
K_sim2 <- envelope(ppm9, Kres, nsim=99, fix.n=T, correction="translation")
G_sim2 <- envelope(ppm9, Gres, nsim=99, fix.n=T, correction="best") lurk1
<- lurking(ppm9, elev, type="Pearson", cumulative=F, envelope=T) par_res
<- parres(ppm9, "elev")
```

This code chunk creates Figure 8.

```
par(mfrow=c(2,2)) plot(K_sim2, xlim=c(0,4000),
legend=F, main="") plot(G_sim2, xlim=c(0,4000),
legend=F, main="")
```

```
plot(lurk1, xlim=c(0,35), main="") plot(par_res,
legend=F, main="")
par(mfrow=c(1,1))
```

Point process modeling of confirmed shell rings

Here we conduct the same analysis as above but on confirmed (i.e., ground-truthed) shell rings. Here we fit a series of point process models to the confirmed shell rings point pattern to assess which variables and parameters best account for the pattern. We begin with a null model (c_ppm0), in this case that the underlying process is complete spatial randomness (CSR), aka a homogeneous Poisson process. We then fit a series of models that contain different combinations of the variables of interest (c_ppm1 through c_ppm7). The default edge correction for ppm is 'border', but here we use 'translation' given interaction distances and anisotropy.

```
c_ppm0 <- ppm(c_rings_pp, ~1, correction="translation") c_ppm1 <-
ppm(c_rings_pp, ~soils_quad, correction="translation" ) c_ppm2 <-
ppm(c_rings_pp, ~elev, correction="translation") c_ppm3 <- ppm(c_rings_pp,
~water_dist, correction="translation") c_ppm4 <- ppm(c_rings_pp,
~soils_quad+elev, correction="translation") c_ppm5 <- ppm(c_rings_pp,
~soils_quad+water_dist, correction="translation") c_ppm6 <- ppm(c_rings_pp,
~elev+water_dist, correction="translation") c_ppm7 <- ppm(c_rings_pp,
~soils_quad+elev+water_dist, correction="translation")
```

We then use AICc and BIC to compare each model.

```
c_MS1_AIC <- model.sel(c_ppm0, c_ppm1, c_ppm2, c_ppm3, c_ppm4, c_ppm5,
c_ppm6, c_ppm7, rank=AICc)
c_MS1_BIC <- model.sel(c_ppm0, c_ppm1, c_ppm2, c_ppm3, c_ppm4, c_ppm5,
c_ppm6, c_ppm7, rank=BIC)
```

Print the model selection results.

```
c_MS1_AIC
```

```
## Model selection table
##        sls_qud     trend df   logLik  AICc delta weight
## c_ppm0                   1 -201.459 405.4  0.00  0.586
## c_ppm2                 e  2 -201.027 407.8  2.35  0.181
## c_ppm3               w_d  2 -201.187 408.1  2.67  0.154
## c_ppm6             e+w_d  3 -199.748 409.5  4.08  0.076
## c_ppm1       +       s_q  4 -200.381 416.8 11.34  0.002
## c_ppm5       +   s_q+w_d  5 -199.382 423.8 18.34  0.000
## c_ppm4       +     s_q+e  5 -200.377 425.8 20.33  0.000
## c_ppm7       + s_q+e+w_d  6 -198.959 437.9 32.50  0.000 ##
Abbreviations:
## trend:  = '~1', e = '~elev', e+w_d = '~elev+water_dist',
##        s_q = '~soils_quad', s_q+e = '~soils_quad+elev',
```

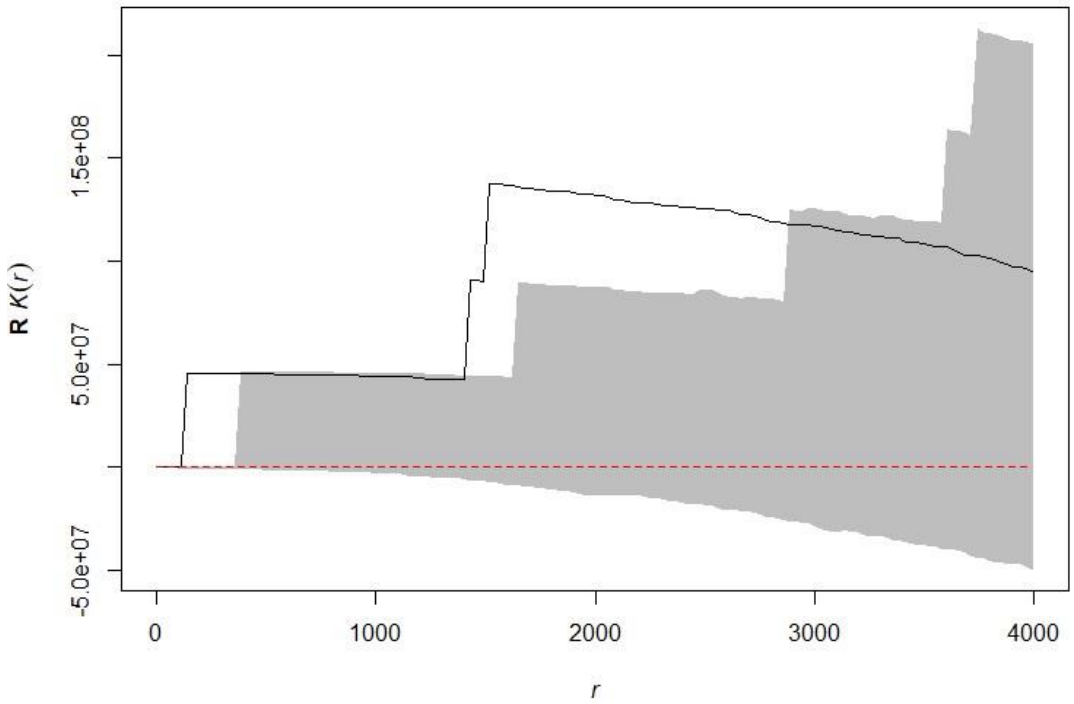
```
##        s_q+e+w_d = '~soils_quad+elev+water_dist',
##        s_q+w_d = '~soils_quad+water_dist', w_d = '~water_dist'
## Models ranked by AICc(x) c_MS1_BIC

## Model selection table
##        sls_qud     trend df   logLik   BIC delta weight
## c_ppm0                   1 -201.459 405.2  0.00  0.371
## c_ppm6             e+w_d  3 -199.748 406.4  1.18  0.205
## c_ppm2                 e  2 -201.027 406.7  1.44  0.181
## c_ppm3               w_d  2 -201.187 407.0  1.76  0.154
## c_ppm1       +       s_q  4 -200.381 410.0  4.75  0.034
## c_ppm5       +   s_q+w_d  5 -199.382 410.3  5.05  0.030
## c_ppm7       + s_q+e+w_d  6 -198.959 411.7  6.51  0.014
## c_ppm4       +     s_q+e  5 -200.377 412.3  7.05  0.011 ##
Abbreviations:
## trend:  = '~1', e = '~elev', e+w_d = '~elev+water_dist',
##        s_q = '~soils_quad', s_q+e = '~soils_quad+elev',
##        s_q+e+w_d = '~soils_quad+elev+water_dist',
##        s_q+w_d = '~soils_quad+water_dist', w_d = '~water_dist'
## Models ranked by BIC(x)
```

Both model selection crieria suggest that the homogeneous Poisson model ppm0 that models the confirmed shell rings as complete spatial randomness (CSR) fits best. However, in both cases the results suggest that models that include elevation have a relatively high ranking, similar to the analysis for all shell rings above. To assess the fit between c_ppm0 and the data, in particular to check for a second-order interaction component, we compare the empirical pattern to expectations from the model by generating 99 simulated realizations of the residual K-function.

```
c_K_sim1 <- envelope(c_ppm0, Kres, nsim=99, fix.n=T,
correction="translation")
```

Plot the residual K-function.

```
plot(c_K_sim1, xlim=c(0,4000), main="", legend=F)
```



The test indicates that the empirical point pattern is more clustered than is accounted for by model c_ppm0. These results are similar to the analysis on all shell rings presented above that show shell rings exhibit a clustered pattern. We then add an interaction term to the model to account for interpoint clustering (c_ppm8). In addition, we also run a model with interpoint clustering and elevation (c_ppm9) Here we use a Gibbs Area Interaction process with irregular parameter r=2000. c_ppm8 only includes the clustering parameter to test the hypothesis that the point pattern is distributed as a series of first-order random clusters, and c_ppm9 evaluates whether the confirmed rings are similar to the total shell rings dataset in being clustered and related to elevation.

```
c_ppm8 <- ppm(c_rings_pp, ~ 1, AreaInter(2000)) c_ppm9
<- ppm(c_rings_pp, ~ elev, AreaInter(2000))
```

We then use AICc and BIC to compare the Area Interaction models (c_ppm8, c_ppm9) with the best-fitting Poisson model (c_ppm0).

```
c_MS2_AIC <- model.sel(c_ppm0, c_ppm8, c_ppm9, rank=AICc) c_MS2_BIC
<- model.sel(c_ppm0, c_ppm8, c_ppm9, rank=BIC)
```

Print model selection results. Both model selection criteria indicate that the Area Interaction model (c_ppm8) fits best. Results suggest that a model with Area Interaction and elevation (c_ppm9) fits better than a model based on complete spatial randomness (c_ppm0).

```
c_MS2_AIC
```

```
## Model selection table
##        trend correction   interaction df   logLik  AICc delta weight
## c_ppm8                  AreIntr(2000)  2 -153.625 313.0  0.00  0.824
## c_ppm9  elev            AreIntr(2000)  3 -153.026 316.1  3.09  0.176
## c_ppm0       translatin                1 -201.459 405.4 92.46  0.000
## Abbreviations:
## trend:  = '~1', elev = '~elev'
## correction: translatin = 'translation'
## interaction: AreIntr(2000) = 'AreaInter(2000)'
## Models ranked by AICc(x) c_MS2_BIC

## Model selection table
##        trend correction   interaction df   logLik   BIC delta weight
## c_ppm8                  AreIntr(2000)  2 -153.625 311.9  0.00  0.635
## c_ppm9  elev            AreIntr(2000)  3 -153.026 313.0  1.11  0.365
## c_ppm0       translatin                1 -201.459 405.2 93.37  0.000
## Abbreviations:
## trend:  = '~1', elev = '~elev'
## correction: translatin = 'translation'
## interaction: AreIntr(2000) = 'AreaInter(2000)'
## Models ranked by BIC(x)
```

Print the parameter estimates for c_ppm8 and standard errors with 95% confidence intervals.

```
coef(summary(c_ppm8))
```

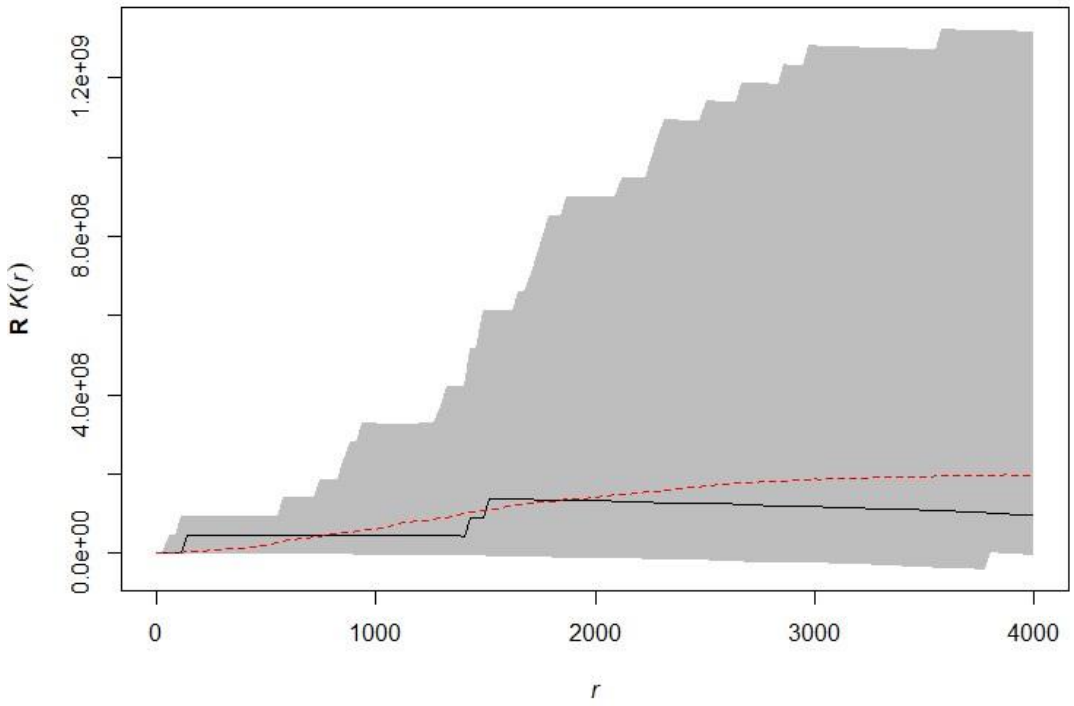
```
##                Estimate      S.E.    CI95.lo    CI95.hi Ztest       Zval
## (Intercept) -19.498567 0.4526883 -20.385820 -18.611314   *** -43.072828
## Interaction   4.187808 0.9439384   2.337723   6.037893   ***   4.436527
```

To assess the fit between the interaction component of this model and the data we generate 99 simulated realizations of the residual K-function.

```
c_K_sim2 <- envelope(c_ppm8, Kres, nsim=99, fix.n=T,
correction="translation", global=F)
```

Plot residual K-function results. Results indicate a good fit between the model (c_ppm8) and the data, suggesting that the confirmed rings are best accounted for as a series of randomly located clusters.

```
plot(c_K_sim2, xlim=c(0,4000), main="", legend=F)
```



Point process modeling of of LiDAR rings

The following code executes the same analyses as above but only for those features identified as 'highly likely' of being shell rings by OBIA of the LiDAR dataset. The results are essentially identical to that of the total dataset.

Run inhomogeneous Poisson models.

```
l_ppm0 <- ppm(l_rings_pp, ~1, correction="translation") l_ppm1 <-
ppm(l_rings_pp, ~soils_quad, correction="translation" ) l_ppm2 <-
ppm(l_rings_pp, ~elev, correction="translation") l_ppm3 <- ppm(l_rings_pp,
~water_dist, correction="translation") l_ppm4 <- ppm(l_rings_pp,
~soils_quad+elev, correction="translation") l_ppm5 <- ppm(l_rings_pp,
~soils_quad+water_dist, correction="translation") l_ppm6 <- ppm(l_rings_pp,
~elev+water_dist, correction="translation") l_ppm7 <- ppm(l_rings_pp,
~soils_quad+elev+water_dist, correction="translation")
```

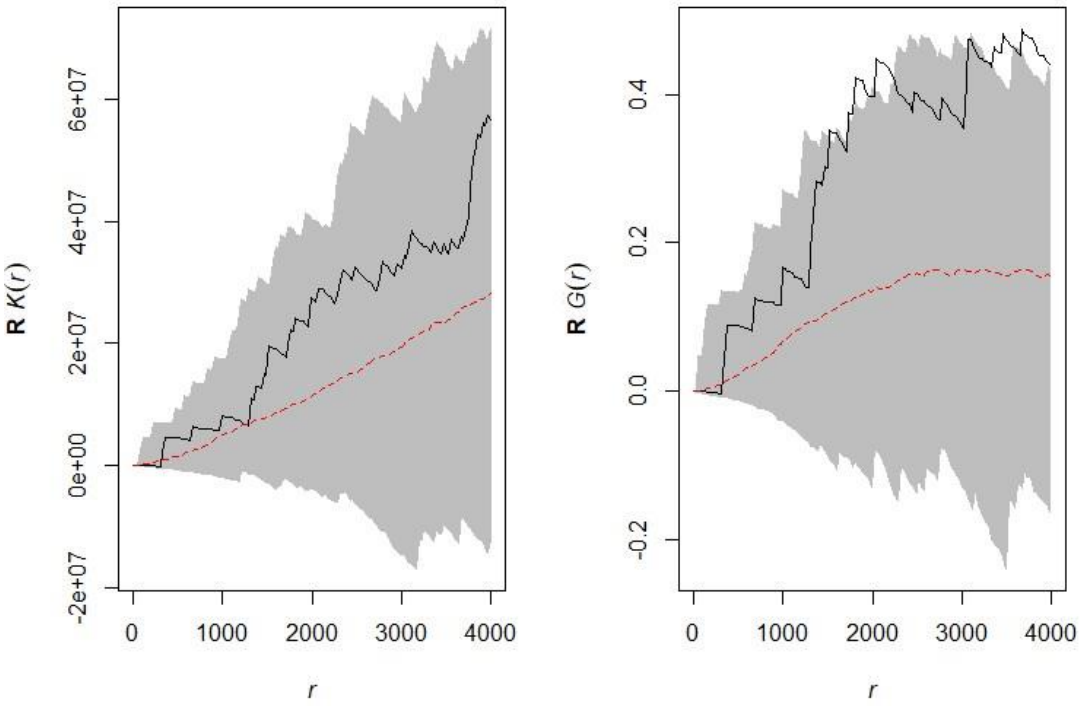
Execute model selection.

```
l_MS1_AIC <- model.sel(l_ppm0, l_ppm1, l_ppm2, l_ppm3, l_ppm4, l_ppm5,
l_ppm6, l_ppm7, rank=AICc)
l_MS1_BIC <- model.sel(l_ppm0, l_ppm1, l_ppm2, l_ppm3, l_ppm4, l_ppm5,
```

```
l_ppm6, l_ppm7, rank=BIC) l_MS1_AIC

## Model selection table
##        sls_qud     trend df   logLik   AICc delta weight
## l_ppm2                 e  2 -770.214 1544.7  0.00  0.696
## l_ppm6             e+w_d  3 -770.202 1547.0  2.30  0.220
## l_ppm4       +     s_q+e  5 -768.898 1549.5  4.73  0.066
## l_ppm7       + s_q+e+w_d  6 -768.893 1552.2  7.45  0.017
## l_ppm1       +       s_q  4 -775.271 1559.6 14.89  0.000
## l_ppm5       +   s_q+w_d  5 -774.201 1560.1 15.33  0.000
## l_ppm3               w_d  2 -782.319 1568.9 24.21  0.000
## l_ppm0                    1 -785.855 1573.8 29.07  0.000 ##
Abbreviations:
## trend:  = '~1', e = '~elev', e+w_d = '~elev+water_dist',
##        s_q = '~soils_quad', s_q+e = '~soils_quad+elev',
##        s_q+e+w_d = '~soils_quad+elev+water_dist',
##        s_q+w_d = '~soils_quad+water_dist', w_d = '~water_dist'
## Models ranked by AICc(x) l_MS1_BIC

## Model selection table
##        sls_qud     trend df   logLik    BIC delta weight
## l_ppm2                 e  2 -770.214 1547.9  0.00  0.853
## l_ppm6             e+w_d  3 -770.202 1551.6  3.71  0.133
## l_ppm4       +     s_q+e  5 -768.898 1556.5  8.58  0.012
## l_ppm7       + s_q+e+w_d  6 -768.893 1560.2 12.31  0.002
## l_ppm1       +       s_q  4 -775.271 1565.5 17.59  0.000
## l_ppm5       +   s_q+w_d  5 -774.201 1567.1 19.19  0.000
## l_ppm3               w_d  2 -782.319 1572.1 24.21  0.000
## l_ppm0                    1 -785.855 1575.4 27.54  0.000 ##
Abbreviations:
## trend:  = '~1', e = '~elev', e+w_d = '~elev+water_dist',
##        s_q = '~soils_quad', s_q+e = '~soils_quad+elev',
##        s_q+e+w_d = '~soils_quad+elev+water_dist',
##        s_q+w_d = '~soils_quad+water_dist', w_d = '~water_dist'
## Models ranked by BIC(x)
```

Run and plot residual K- and G-functions.

```
l_K_sim1 <- envelope(l_ppm2, Kres, nsim=99, fix.n=T,
correction="translation")
l_G_sim1 <- envelope(l_ppm2, Gres, nsim=99, fix.n=T, correction="best")
par(mfrow=c(1,2)) plot(l_K_sim1, xlim=c(0,4000), main="", legend=F)
plot(l_G_sim1, xlim=c(0,4000), main="",legend=F)
```




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

Run Area Interaction models.

```
l_ppm8 <- ppm(l_rings_pp, ~ 1, AreaInter(2000)) l_ppm9
<- ppm(l_rings_pp, ~ elev, AreaInter(2000))
```

Run model selection for Area Interaction models

```
l_MS2_AIC <- model.sel(l_ppm0, l_ppm8, l_ppm9, rank=AICc)
l_MS2_BIC <- model.sel(l_ppm0, l_ppm8, l_ppm9, rank=BIC) l_MS2_AIC

## Model selection table
##        trend correction   interaction df   logLik   AICc  delta weight
## l_ppm9  elev            AreIntr(2000)  3 -540.699 1088.0   0.00  0.993
## l_ppm8                  AreIntr(2000)  2 -546.747 1097.8   9.77  0.007
## l_ppm0       translatin                1 -785.855 1573.8 485.78  0.000
## Abbreviations:
## trend:  = '~1', elev = '~elev'
## correction: translatin = 'translation'
## interaction: AreIntr(2000) = 'AreaInter(2000)'
## Models ranked by AICc(x) l_MS2_BIC



## Model selection table
```

```
##        trend correction   interaction df   logLik    BIC  delta weight
## l_ppm9  elev            AreIntr(2000)  3 -540.699 1092.6   0.00  0.985
## l_ppm8                  AreIntr(2000)  2 -546.747 1101.0   8.36  0.015
## l_ppm0       translatin                1 -785.855 1575.4 482.84  0.000
## Abbreviations:
## trend:  = '~1', elev = '~elev'
## correction: translatin = 'translation'
## interaction: AreIntr(2000) = 'AreaInter(2000)'
## Models ranked by BIC(x)
```

Print the parameter estimates for l_ppm9 and standard errors with 95% confidence intervals.

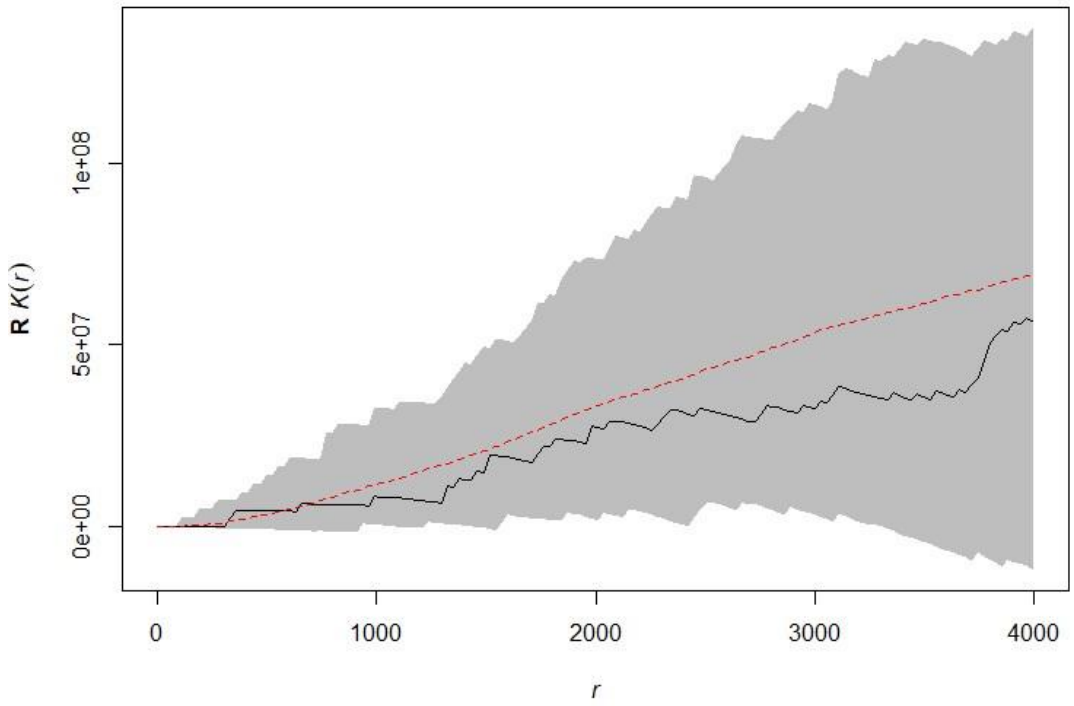
```
coef(summary(l_ppm9))
```

```
##                 Estimate       S.E.      CI95.lo      CI95.hi Ztest
## (Intercept) -18.75437448 0.34197235 -19.42462797 -18.08412100   ***
## elev          0.05952516 0.01969005   0.02093338   0.09811694    **
## Interaction   1.53361726 0.94050911  -0.30974672   3.37698124      ##
Zval
## (Intercept) -54.841786
## elev          3.023109
## Interaction   1.630625
```

Plot residual K-function results.

```
l_K_sim2 <- envelope(l_ppm9, Kres, nsim=99, fix.n=T,
correction="translation", global=F)
```

```
## Generating 99 simulated realisations of fitted Gibbs model with fixed  ##
number of points  ...
## 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(l_K_sim2, xlim=c(0,4000),
main="", legend=F)
```


R K(r)
0e+00
5e+07
1e+08
0
1000
2000
3000
4000
r