TASK 3: Interaction

Ferrara Lorenzo

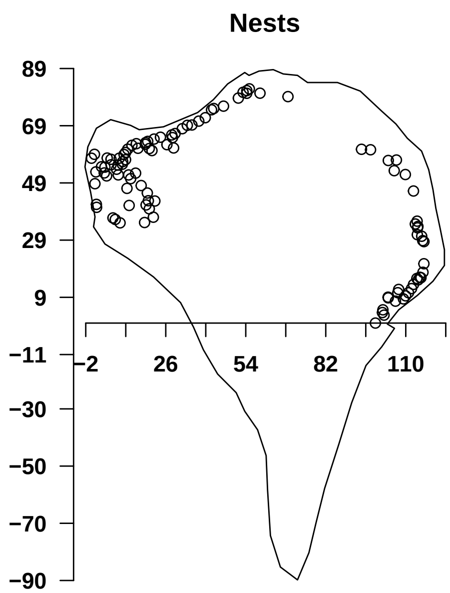
Task 3: Interaction

The nest data from islet "nucli 84" is stored in nucli84.txt. Additionally, the coordinates of the islet are in poly84.txt

1. Build a ppp object using the "nucli 84" data.

```
rm(list = ls())
nucli84 <- read.delim("T3/nucli84.txt")

min.X = min(nucli84$X)
min.Y = min(nucli84$Y)
max.X = max(nucli84$X)
max.Y = max(nucli84$Y)

nucli84$X = nucli84$X - min.X
nucli84$Y = nucli84$Y - min.Y

n84 = ppp(x = nucli84$X, y = nucli84$Y, range(nucli84$X), range(nucli84$Y))

poligon = read.delim("T3/poly84.txt")

poligon$X = poligon$X - min.X
poligon$Y = poligon$Y - min.Y
pol.illa <- list(x = poligon$X, y = poligon$Y)

min.pX = min(poligon$X)
min.pY = min(poligon$Y)
max.pX = max(poligon$X)
max.pY = max(poligon$Y)

n84p = ppp(nucli84$X, nucli84$Y, poly = pol.illa, range(poligon$X), range(poligon$Y))
```

```
par(mfrow = c(1, 1), font = 2, font.axis = 2, font.lab = 4, las = 1, mar = c(0, 0,
    2, 0))
plot(n84p, main = "Nests")
axis(1, at = c(round(seq(min.pX, max.pX, length = 10), digits = 0)), pos = c(0, 0))
axis(2, at = c(round(seq(min.pY, max.pY, length = 10), digits = 0)), pos = c(min.pX -
    4, min.pY - 50))
```



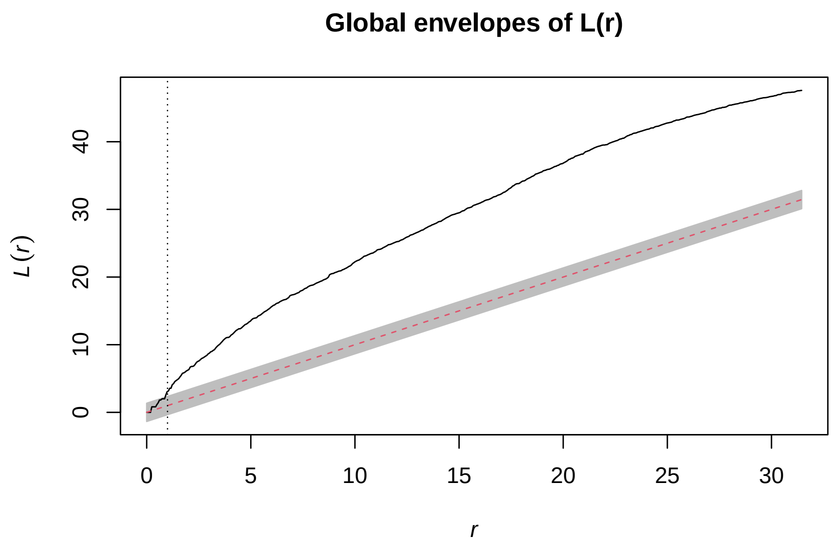
2) Check the interaction pattern.

To check the interaction pattern, we utilize the L-function defined as $L(r) = \sqrt{\frac{K(r)}{\pi}}$ as it is an estimator with a variance that does not vary with respect to r. This makes it more stable than other estimators, such as F, G, and K.

```
E <- envelope(n84p, Lest, nsim = 19, rank = 1, global = TRUE, correction = "best")
```

```
## Generating 19 simulations of CSR  ...
## 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18,  19.
##
## Done.
```

```
plot(E, main = "Global envelopes of L(r)", legend = F)
abline(v = 1, lty = 3)
```



The observed $L(r)$ goes above the theoretical function, indicating that the observed distance between locations is greater than expected under the assumption of a homogeneous Poisson process. This suggests the presence of a regular pattern. Additionally, we notice that the observed line falls outside of the envelopes from $r = 1$ onward.

3) Model the relation between the intensity of the point process and the covariates height and vegetation accounting for the interaction pattern. Interpret the estimates.

Since we have a regular pattern, we will use Gibbs models.

First we try to fit a Hardcore model:

```
grid <- read.delim("T2/grid.txt")
grid.veg = read.delim("T2/grid_veg.txt")
veg = as.matrix(read.delim("T2/veg.txt", header = FALSE))
height = as.matrix(read.delim("T2/height.txt", header = FALSE))

Height = im(mat = height, xcol = grid$x, yrow = grid$y)
Vegetation = im(mat = veg, xcol = grid.veg$x, yrow = grid.veg$y)

fitH <- ppm(n84p, ~Height + Vegetation, Hardcore)
```

```
summary(fitH)
```

```
## Point process model
## Fitting method: maximum pseudolikelihood (Berman-Turner approximation)
## Model was fitted using glm()
## Algorithm converged
## Call:
## ppm.ppp(Q = n84p, trend = ~Height + Vegetation, interaction = Hardcore)
## Edge correction: "border"
##  [border correction distance r = 0.241749245771188 ]
```

```
## ------------------------------------------------------------------------------
## Quadrature scheme (Berman-Turner) = data + dummy + weights
##
## Data pattern:
## Planar point pattern:  104 points
## Average intensity 0.00926 points per square unit
## Window: polygonal boundary
## single connected closed polygon with 60 vertices
## enclosing rectangle: [-2.26699, 123.5272] x [-89.78583, 88.59802] units
##                      (125.8 x 178.4 units)
## Window area = 11231.3 square units
## Fraction of frame area: 0.501
##
## Dummy quadrature points:
##      32 x 32 grid of dummy points, plus 4 corner points
##      dummy spacing: 3.931068 x 5.574495 units
##
## Original dummy parameters: =
## Planar point pattern:  560 points
## Average intensity 0.0499 points per square unit
## Window: polygonal boundary
## single connected closed polygon with 60 vertices
## enclosing rectangle: [-2.26699, 123.5272] x [-89.78583, 88.59802] units
##                      (125.8 x 178.4 units)
## Window area = 11231.3 square units
## Fraction of frame area: 0.501
## Quadrature weights:
##      (counting weights based on 32 x 32 array of rectangular tiles)
## All weights:
##  range: [0.71, 21.9] total: 11200
## Weights on data points:
##  range: [3.1, 11]    total: 711
## Weights on dummy points:
##  range: [0.71, 21.9] total: 10500
## ------------------------------------------------------------------------------
## FITTED :
##
## Nonstationary Hard core process
##
## ---- Trend: ----
##
## Log trend: ~Height + Vegetation
## Model depends on external covariates 'Height' and 'Vegetation'
## Covariates provided:
##  Height: im
##  Vegetation: im
##
## Fitted trend coefficients:
## (Intercept)      Height  Vegetation
## -7.54677892  0.15928955  0.01532893
##
##                Estimate       S.E.      CI95.lo     CI95.hi Ztest      Zval
## (Intercept) -7.54677892 0.75595463 -9.028422773 -6.06513506   *** -9.983111
## Height       0.15928955 0.01722086  0.125537275  0.19304182   ***  9.249800
```

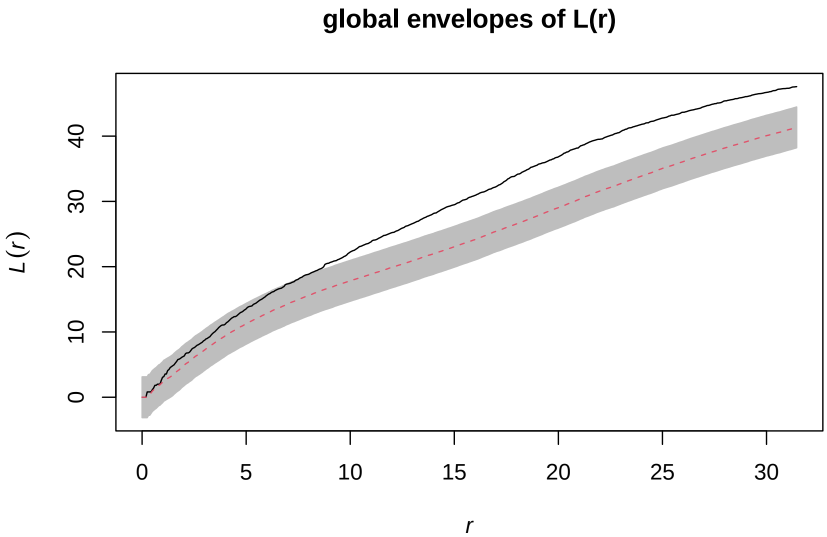
```
## Vegetation   0.01532893 0.01097206 -0.006175909  0.03683378        1.397088
##
##  ---- Interaction: -----
##
## Interaction: Hard core process
## Hard core distance:  0.2417492
##
## ----------- gory details -----
##
## Fitted regular parameters (theta):
## (Intercept)      Height  Vegetation
## -7.54677892  0.15928955  0.01532893
##
## Fitted exp(theta):
##  (Intercept)       Height   Vegetation
## 0.0005278075 1.1726774457 1.0154470245
```

We assess the goodness of fit of the model by looking at the theoretical envelopes of $L(r)$ and comparing it with the observed values.

```
E <- envelope(fitH, Lest, nsim = 19, rank = 1, global = TRUE, correction = "best")
```

```
## Generating 38 simulated realisations of fitted Gibbs model (19 to estimate the
## mean and 19 to calculate envelopes) ...
## 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
##
## Done.
```

```
plot(E, main = "global envelopes of L(r)", legend = F)
```



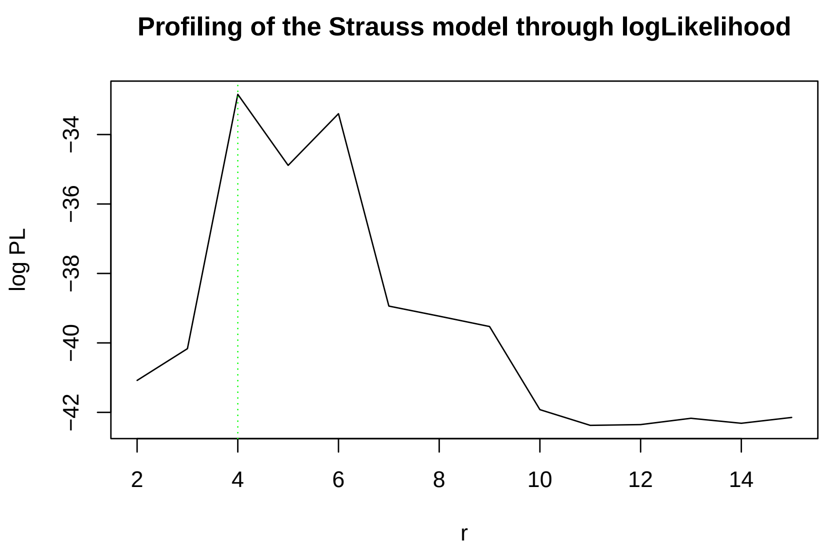
The observed function falls outside of the envelopes, indicating that the Hardcore model does not fit the data well.

As an alternative, we will try using a Strauss model. Our exploratory analysis showed that the observed L-function lies outside the envelope for values of r greater than 1. To determine the optimal value for r, we will analyze the profile likelihood using a range of values inside the interval $[1, 15]$ and choose the one that provides the best fit to the data.

```
df = data.frame(r = seq(1, 15, by = 0.5))
pfit = profilepl(df, Strauss, n84p, ~Height + Vegetation)
```

```
## 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
```

```
plot(pfit, main = "Profiling of the Strauss model through logLikelihood")
```




```
r.opt = pfit$fit$fitin$interaction$par$r
```

The optimal value is reached at $r = 4$.

So now we can finally estimate the model:

```
fitSt <- ppm(n84p, ~Height + Vegetation, Strauss(r = r.opt))
fitSt
```

```
## Nonstationary Strauss process
##
## Log trend:  ~Height + Vegetation
##
## Fitted trend coefficients:
## (Intercept)      Height  Vegetation
## -8.50843012  0.10802394  0.03107911
##
## Interaction distance:    4
## Fitted interaction parameter gamma:  1.5200358
##
```

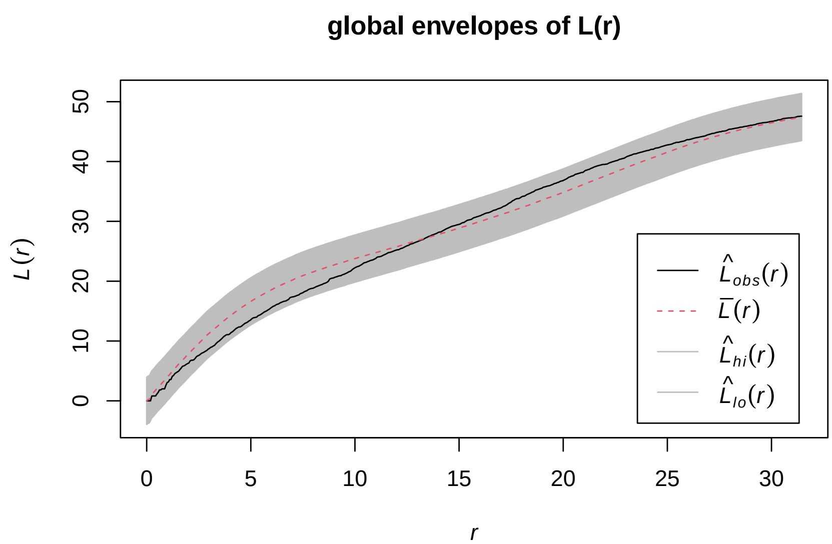
```
## Relevant coefficients:
## Interaction
##   0.4187339
##
## For standard errors, type coef(summary(x))
##
## *** Model is not valid ***
## *** Interaction parameters are outside valid range ***
```

and assess his validity:

```r
E <- envelope(fitSt, Lest, nsim = 19, rank = 1, global = TRUE, correction = "best")
```

```
## Generating 38 simulated realisations of fitted Gibbs model (19 to estimate the
## mean and 19 to calculate envelopes) ...
## 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
##
## Done.
```

```r
plot(E, main = "global envelopes of L(r)")
```



The observed L-function is inside the envelopes, therefore the Strauss model can be valid and we proceed to interpret it.

```r
summary(fitSt, fine = F)
```

```
## Point process model
## Fitting method: maximum pseudolikelihood (Berman-Turner approximation)
## Model was fitted using glm()
## Algorithm converged
## Call:
```

```
## ppm.ppp(Q = n84p, trend = ~Height + Vegetation, interaction = Strauss(r = r.opt))
## Edge correction: "border"
##  [border correction distance r = 4 ]
## --------------------------------------------------------------------------------
## Quadrature scheme (Berman-Turner) = data + dummy + weights
##
## Data pattern:
## Planar point pattern:  104 points
## Average intensity 0.00926 points per square unit
## Window: polygonal boundary
## single connected closed polygon with 60 vertices
## enclosing rectangle: [-2.26699, 123.5272] x [-89.78583, 88.59802] units
##                      (125.8 x 178.4 units)
## Window area = 11231.3 square units
## Fraction of frame area: 0.501
##
## Dummy quadrature points:
##      32 x 32 grid of dummy points, plus 4 corner points
##      dummy spacing: 3.931068 x 5.574495 units
##
## Original dummy parameters: =
## Planar point pattern:  560 points
## Average intensity 0.0499 points per square unit
## Window: polygonal boundary
## single connected closed polygon with 60 vertices
## enclosing rectangle: [-2.26699, 123.5272] x [-89.78583, 88.59802] units
##                      (125.8 x 178.4 units)
## Window area = 11231.3 square units
## Fraction of frame area: 0.501
## Quadrature weights:
##      (counting weights based on 32 x 32 array of rectangular tiles)
## All weights:
##  range: [0.71, 21.9] total: 11200
## Weights on data points:
##  range: [3.1, 11]    total: 711
## Weights on dummy points:
##  range: [0.71, 21.9] total: 10500
## --------------------------------------------------------------------------------
## FITTED :
##
## Nonstationary Strauss process
##
## ---- Trend: ----
##
## Log trend: ~Height + Vegetation
## Model depends on external covariates 'Height' and 'Vegetation'
## Covariates provided:
##  Height: im
##  Vegetation: im
##
## Fitted trend coefficients:
## (Intercept)      Height  Vegetation
## -8.50843012  0.10802394  0.03107911
##
```

```
##                Estimate       S.E.       CI95.lo     CI95.hi Ztest       Zval
## (Intercept) -8.50843012 0.84655940 -10.167656050 -6.84920419   *** -10.050600
## Height       0.10802394 0.01884424   0.071089902  0.14495798   ***   5.732463
## Vegetation   0.03107911 0.01163310   0.008278644  0.05387957    **   2.671609
## Interaction  0.41873389 0.09133454   0.239721476  0.59774630   ***   4.584617
##
##  ---- Interaction: -----
##
## Interaction: Strauss process
## Interaction distance:    4
## Fitted interaction parameter gamma:  1.5200358
##
## Relevant coefficients:
## Interaction
##   0.4187339
##
## ----------- gory details -----
##
## Fitted regular parameters (theta):
## (Intercept)      Height  Vegetation Interaction
## -8.50843012  0.10802394  0.03107911  0.41873389
##
## Fitted exp(theta):
##  (Intercept)       Height   Vegetation  Interaction
## 0.0002017603 1.1140744182 1.0315671056 1.5200358028
##
## *** Model is not valid ***
## *** Interaction parameters are outside valid range ***
```

The estimates of its coefficients are:

```
fitSt$coef
```

```
## (Intercept)      Height  Vegetation Interaction
## -8.50843012  0.10802394  0.03107911  0.41873389
```

The estimate for the interaction parameter is $\gamma = 0.4187339$. The parameter γ controls the strength of interaction between points. If $\gamma = 1$ the model reduces to a Poisson process. If $\gamma = 0$ the model is a hard core process. For values $0 < \gamma < 1$, like in our case, the process exhibits a moderate strength of interaction between the presence of points in different locations.