GET: Hotspot detection on a linear network

Mari Myllymäki
Natural Resources Institute Finland (Luke)

Tomáš Mrkvička
University of South Bohemia

Abstract

This vignette shows how the methodology proposed by Mrkvička, Kraft, Blažek, and Myllymäki (2023) for detecting hotspots on a linear network can be performed using the R package **GET** (Myllymäki and Mrkvička 2023).

Keywords: false discovery rate, hotspot, linear network, Monte Carlo test, road accidents, R, spatial point pattern.

Loading required packages and setting a **ggplot2** theme for images.

```
R> library("GET")
R> library("spatstat")
R> library("spatstat.linnet")
R> #library("rgdal")
R> #library("raster")
R> #library("tiff")
R> #library("imager")
R> #library("maptools")
R> library("ggplot2")
R> theme_set(theme_bw(base_size = 9))
```

1. Data

Mrkvička *et al.* (2023) worked with the database of road crashes reported to the Police in the Czech Republic from 1 January 2016 to 31 December 2020. Here we show the methodology for a subpattern of this full data set. The **GET** package provides a data object `roadcrash` that has 7700 road crashes lying on a linear network with 269 vertices and 354 lines. Because the computations of inhomogeneous K-function and density are rather computational (functions `linearKinhom()` and `density.lpp()` of the **spatstat** package, Baddeley, Rubak, and Turner 2015), for illustration of the methodology below, we use a subset of the `roadcrash` data.

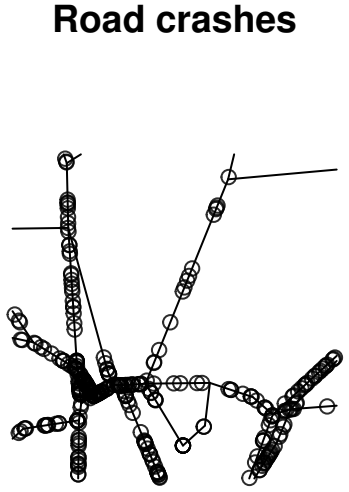
Load the road crash data from **GET**:

```
R> data("roadcrash")
R> win <- owin(xrange = roadcrash$xrange,
+              yrange = roadcrash$yrange)
R> X <- ppp(x = roadcrash$x, y = roadcrash$y, window = win)
R> Vertices.pp <- ppp(x = roadcrash$Vertices.x,
```

```
+                     y = roadcrash$Vertices.y,
+                     window=win)
R> L <- linnet(vertices=Vertices.pp,
+              edges = roadcrash$Edges)
R> PPfull <- lpp(X, L)
R> roadcrash$Traffic <- im(roadcrash$Traffic,
+           xrange = roadcrash$xrange,
+           yrange = roadcrash$yrange)
R> roadcrash$ForestDensity <- im(roadcrash$ForestDensity,
+           xrange = roadcrash$xrange,
+           yrange = roadcrash$yrange)
R> roadcrash$BuildingDensity <- im(roadcrash$BuildingDensity,
+           xrange = roadcrash$xrange,
+           yrange = roadcrash$yrange)
```

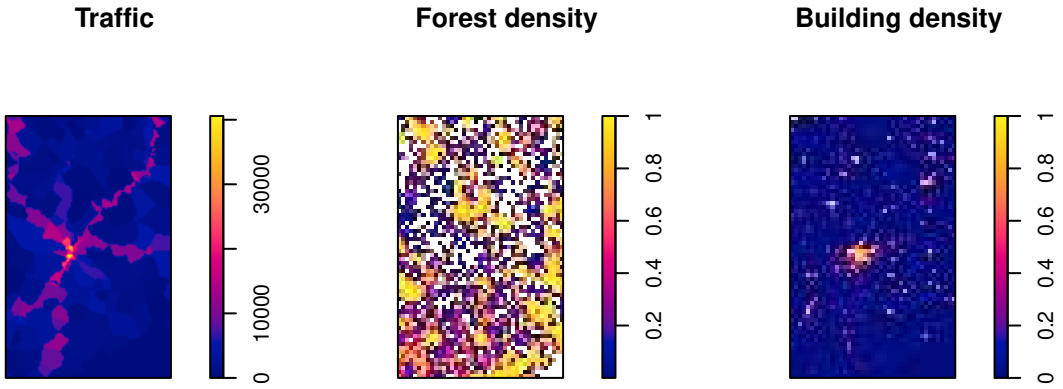
Define a subpattern:

```
R> subwin <- owin(c(-760000, -750000), c(-1160000, -1150000))
R> PP <- PPfull[, subwin]
R> plot(PP, main="Road crashes")
```

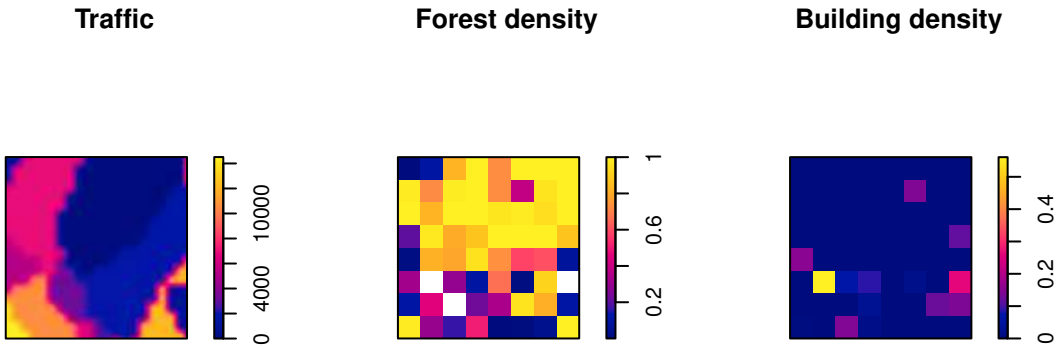


Mrkvička *et al.* (2023) had a total of 9 spatially defined covariates. In our example here and available in `roadcrash` in **GET** are three covariates, namely average traffic volume (number of vehicles per 24 hours), forest density and building density in the cell.

```
R> par(mfrow=c(1,3))
R> plot(roadcrash$Traffic, main="Traffic")
R> plot(roadcrash$ForestDensity, main="Forest density")
R> plot(roadcrash$BuildingDensity, main="Building density")
```




```
R> par(mfrow=c(1,3))
R> plot(roadcrash$Traffic[subwin], main="Traffic")
R> plot(roadcrash$ForestDensity[subwin], main="Forest density")
R> plot(roadcrash$BuildingDensity[subwin], main="Building density")
```

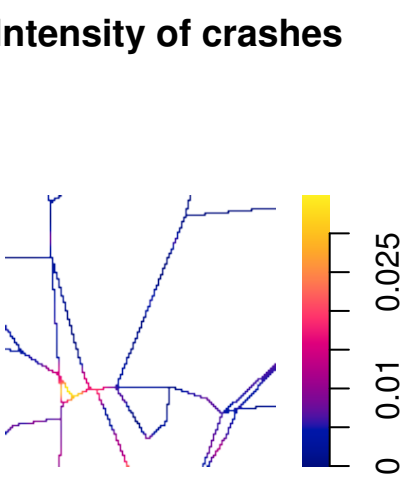


2. Non-parametric intensity estimate

A non-parametric density estimate of the point pattern on a linear network can be obtained

using the function `density.lpp()` of the **spatstat** package.

```
R> densi <- density.lpp(PP, sigma = 250, distance="euclidean")
R> plot(densi, main="Intensity of crashes")
```



3. Fitting the Matern cluster process on a linear network

The simplest point process model for road crashes is the (inhomogeneous) Poisson process with intensity

$$\rho_\beta(u) = \kappa \exp(z(u)\beta^T), \ u \in L, \tag{1}$$

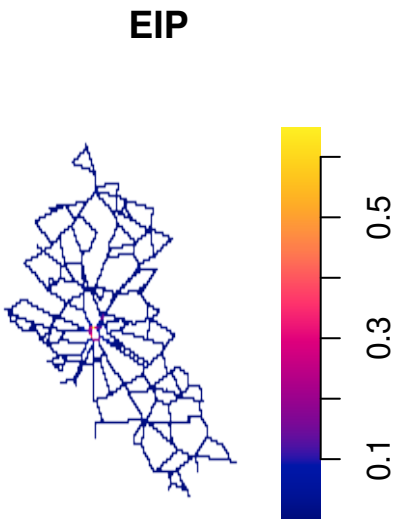
where L is a linear network, $z = (z_1, \ldots, z_k)$ is a vector of covariates and $\beta = (\beta_1, \ldots, \beta_k)$ is a regression parameter. This process can be fitted using the **spatstat** package. We fit the model using the full `roadcrash` data.

```
R> M1 <- lppm(PPfull ~ Traffic + ForestDensity + BuildingDensity, data = roadcrash)
R> M1

Point process model on linear network
        Fitted to point pattern dataset 'PPfull'

Nonstationary Poisson process

Log intensity:  ~Traffic + ForestDensity + BuildingDensity
```

```
Fitted trend coefficients:
    (Intercept)         Traffic   ForestDensity BuildingDensity
  -5.887298e+00    9.614399e-05   -1.021966e-01    2.307020e+00

                     Estimate         S.E.       CI95.lo
(Intercept)     -5.887298e+00 2.457105e-02 -5.935456e+00
Traffic          9.614399e-05 1.257982e-06  9.367839e-05
ForestDensity   -1.021966e-01 4.518626e-02 -1.907601e-01
BuildingDensity  2.307020e+00 3.219970e-02  2.243910e+00
                      CI95.hi Ztest        Zval
(Intercept)     -5.839139e+00   *** -239.602974
Traffic          9.860959e-05   ***   76.427190
ForestDensity   -1.363316e-02     *   -2.261674
BuildingDensity  2.370130e+00   ***   71.647238

Domain: Linear network with 269 vertices and 354 lines
Enclosing window: rectangle = [-774936.9, -727048.9] x [-1201599.8,
-1125679.8] units
```

The predicted point process intensity can then be obtained by the `predict()` function.

```
R> EIP <- predict(M1)
R> plot(EIP)
```



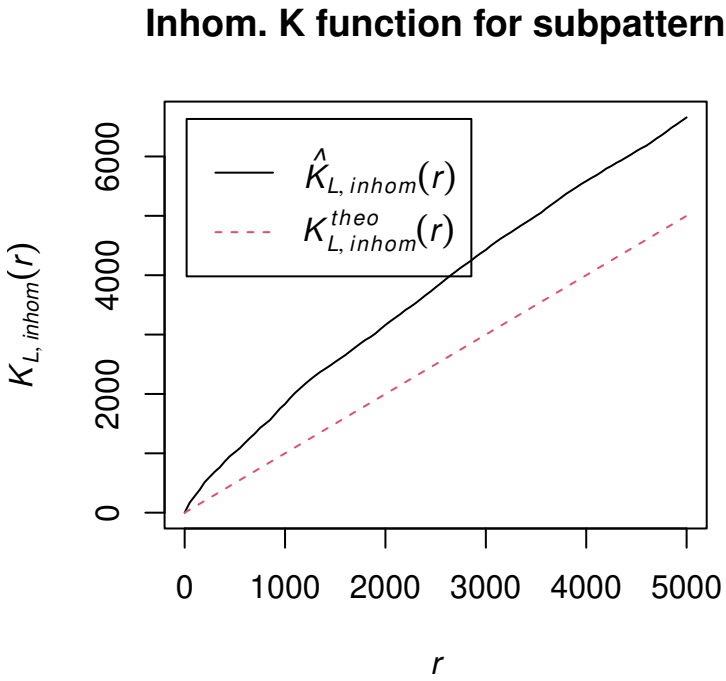
Mrkvička *et al.* (2023) considered instead of the Poisson process the Matern cluster point process with inhomogeneous cluster centers. This process is more suitable for clustered data. It can be estimated in two steps according to its construction following Mrkvička, Muška, and Kubečka (2014). In first step, the first order intensity function is estimated through Poisson likelihood. This was done above, i.e., the object `EIP` contains the estimated intensity. In second step, the second order interaction parameters α (mean number of points in a cluster) and R (cluster radius) are estimated through minimum contrast method. Unfortunately, working with cluster processes on linear networks is very time consuming and therefore they are currently not covered by the **spatstat** package. Thus, we have used the inhomogeneous K-function and the minimum contrast and grid search methods to find the optimal parameters as follows:

First, a function to simulate the Matern cluster process on a linear network `LL` with pre-specified centers:

```
R> # Matern cluster process on a linear network
R> # Centers = (x,y)-coordinates of parent points
R> # R = The R parameter of the Matern cluster process
R> # alpha =  The alpha parameter of the Matern cluster process
R> # LL = The linear network on which the point pattern should be simulated.
R> rMatClustlpp <- function(Centers, R, alpha, LL) {
+    X <- array(0,0)
+    Y <- array(0,0)
+    for(p in 1:length(Centers$data$x)) {
+      BBCOutD <- disc(radius=R, centre=c(Centers$data$x[p],
+                                         Centers$data$y[p]),
+                      npoly = 32)
+      BBCD <- intersect.owin(LL$window, BBCOutD)
+      if(volume(LL[BBCOutD])>0) {
+        Xp <- rpoislpp(alpha/volume(LL[BBCOutD]), L=LL[BBCD])
+        X <- append(X, as.numeric(Xp$data$x))
+        Y <- append(Y, as.numeric(Xp$data$y))
+      }
+    }
+    lpp(cbind(X,Y), LL)
+  }
```

Then we estimate the parameters R and α of the Matern cluster process using the inhomogeneous K-function with the estimated Poisson process intensity from the model `M1` fitted above. First we estimate the K-function for the data pattern (subpattern of road crash data):

```
R> r <- seq(0, 5000, by=50) # Up to 5 km
R> # Takes about 3 minutes for the 7700 crashes; 4 sec for thinned pattern
R> PP_K <- linearKinhom(PP, lambda = M1, r=r)
R> plot(PP_K, main = "Inhom. K function for subpattern")
```



Then we estimate of Matern cluster process parameters. Because the theoretical K-function is not known for this process on the linear network, we approximate it from `nsim` simulations from the process. And, we consider a range of possible values of the parameters R and α.

```
R> nsim <- 10
R> valpha <- seq(5, 30, by=5)
R> vR <- seq(250, 2500, by=500)
```

For each value of R and α, we compute the difference of the observed K-function from the "theoretical" K-function of the model, computed from the average of `nsim` simulation from the model. (Note that this computation takes some time; for the subpattern with 337 points and above parameter values, this took using a single core on a laptop about 11 min.)

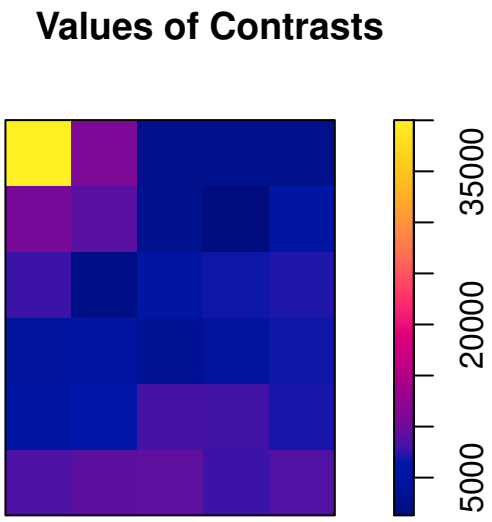
```
R> set.seed(2023)
```

```
R> Contrast <- array(0, c(length(valpha), length(vR)))
R> for(i in 1:length(valpha)) {
+    for(j in 1:length(vR)) {
+      # Compute the average K of 10 simulation from the model
+      KMC <- array(0,length(r))
+      for(s in 1:nsim) {
+        # Centers from a Poisson process
+        Centers <- rpoislpp(EIP/valpha[i], L=PP[['domain']])
+        XX <- rMatClustlpp(Centers, vR[j], valpha[i], PP[['domain']])
+        KMC <- KMC + linearKinhom(XX, lambda = M1, r=r)$est
```

```
+      }
+      # Compute the difference between estimated and average K
+      Contrast[i,j] <- sqrt(sum((PP_K$est-KMC/nsim)^2))
+    }
+  }
```

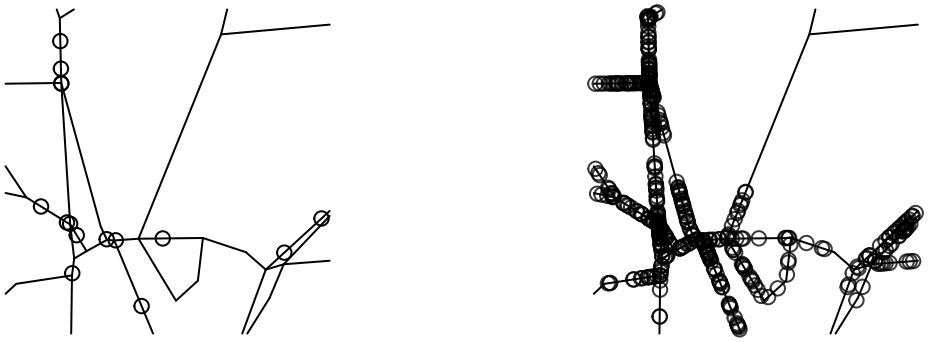
We then find out which of these possible values of parameters R and α lead to the smallest difference between the observed and "theoretical" K-functions.

```
R> plot(as.im(Contrast), main="Values of Contrasts")
R> # Finding the minimum value of the contrast
R> id <- which(Contrast == min(Contrast), arr.ind = TRUE)
R> alpha <- valpha[id[,1]]
R> R <- vR[id[,2]]
R> # Chosen values
R> alpha

[1] 25

R> R

[1] 1750
```



A simulation from the fitted Matern cluster process can be generated as follows:

```
R> # Centers from a Poisson process
R> Centers <- rpoislpp(EIP/alpha, L=PP[['domain']])
R> par(mfrow=c(1,2))
R> plot(Centers)
R> XX <- rMatClustlpp(Centers, R, alpha, PP[['domain']])
R> plot(XX, main="A realization of the model")
```



In what follows, we utilize these simulations from the fitted model in order to construct the envelope representing the behavior of the intensity under the fitted model and to find the hotspots that are not explained by the three covariates.

4. False discovery rate envelopes

To find the hotspots of road crashes that are not explained by the covariates, we first generate `nsim` simulations from the fitted Matern cluster process and estimate the intensity for each of the simulated patterns. Note that the intensity of the observed pattern was estimated similarly for the observed pattern above. (Note: This takes a lot of time; 10000 simulations took about 3 hours for the subpattern with 337 points.)

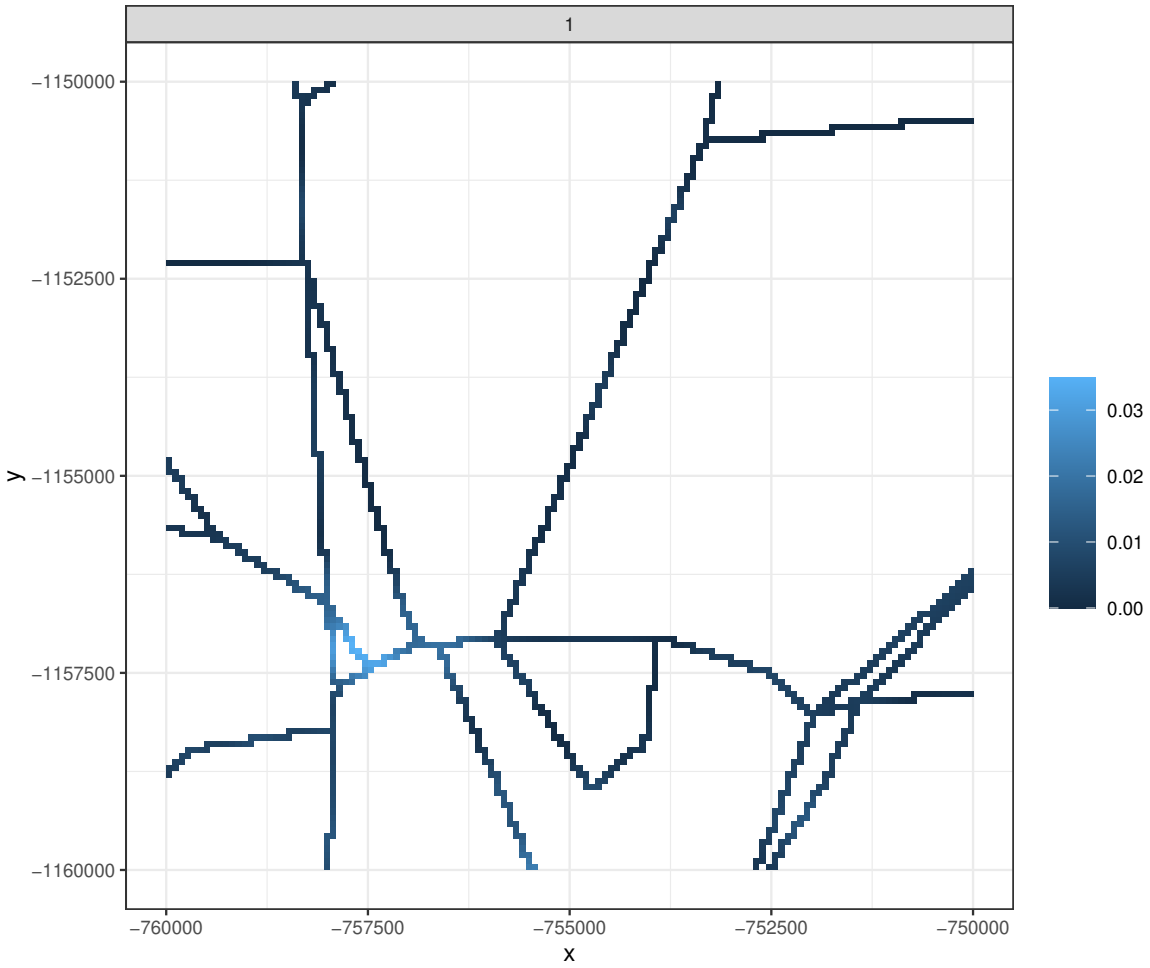
```
R> nsim <- 10000
R> sims.densi <- vector(mode = "list", length = nsim)
R> for(s in 1:nsim) {
+    Centers <- rpoislpp(EIP/alpha, L=PP[['domain']])
+    simss <- rMatClustlpp(Centers, R, alpha, PP[['domain']])
+    sims.densi[[s]] <- density.lpp(simss, sigma = 250, distance="euclidean")
+  }
```

Before computing the FDR envelope, as an additional task, we need to take care of the NA values outside the network. This is done by finding where the NAs in the observed and simulated intensities occur, and preparing a `curve_set` object containing all the non-NA values. In the `curve_set` object, `r` specifies the observation locations, `obs` is the observed intensity values and `sim_m` contains the simulated intensities.

```
R> yx <- expand.grid(densi$yrow, densi$xcol)
R> noNA_id <- which(!is.na(densi$v))
R> noNA_idsim <- which(!is.na(sims.densi[[1]]$v))
R> noNA_id <- intersect(noNA_id, noNA_idsim)
R> #max(densi[noNA_id]); summary(sapply(sims.densi, FUN=max))
R>
R> cset <- create_curve_set(list(
+    r=data.frame(x=yx[,2], y=yx[,1],
+                 width=densi$xstep, height=densi$ystep)[noNA_id,],
+    obs=as.vector(densi$v)[noNA_id],
+    sim_m=sapply(sims.densi, FUN=function(x){ as.vector(x$v)[noNA_id] },
+                 simplify=TRUE)))

R> #save(cset, file="roadcrash_cset.Rdata")
R> load(file="roadcrash_cset.Rdata")

R> cset

A curve_set(2d) object with 10001 curves observed at 920 argument values
(1 observed, 10000 simulated).
Contains:
$ r     : 'data.frame':        920 obs. of  4 variables:
 $ x     : num  -759961 -759961 -759961 -759961 -759961 ...
 $ y     : num  -1158789 -1158711 -1155664 -1154961 -1154883 ...
 $ width : num  78.1 78.1 78.1 78.1 78.1 ...
```

```
 $ height: num  78.1 78.1 78.1 78.1 78.1 ...
$ funcs :  num [1:920, 1:10001] 0.00452 0.00515 0.00307 0.00513 0.005 ...
 - attr(*, "dimnames")=List of 2
  ..$ : NULL
  ..$ : chr [1:10001] "obs" "sim1" "sim2" "sim3" ...
```

```
R> plot(cset, idx=1, main="The observed intensity")
```

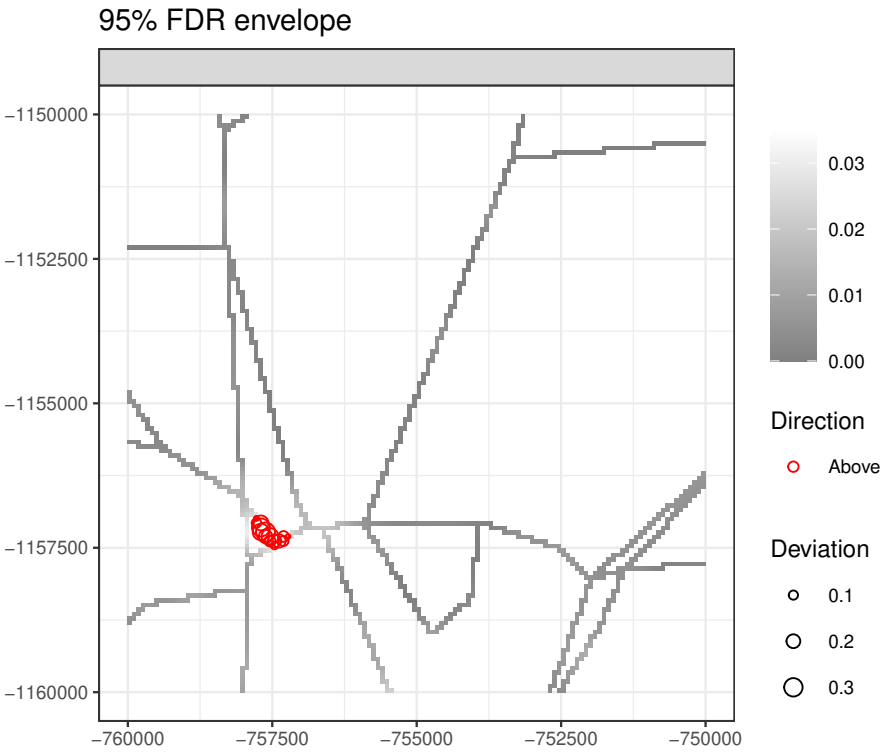


Then the FDR envelope (Mrkvička and Myllymäki 2023) can be computed using the function `fdr_envelope()` of the **GET** package. We set the alternative to `"greater"`, because we are only interested in locations where the intensity is higher than expected.

```
R> res <- fdr_envelope(cset, alternative = "greater")
R> res
```

```
FDR envelope based on simulations:
 Number of rejected hypotheses: 15
 Number of accepted hypotheses: -15
 Total number of hypotheses: 0
```

```
R> plot(res) + scale_radius(range = 0.5 * c(1, 6))
```



References

Baddeley A, Rubak E, Turner R (2015). *Spatial Point Patterns: Methodology and Applications with R*. Chapman and Hall/CRC Press, London.

Mrkvička T, Kraft S, Blažek V, Myllymäki M (2023). "Hotspot Detection on a Linear Network in the Presence of Covariates: A Case Study on Road Crash Data." doi:http://dx.doi.org/10.2139/ssrn.4627591.

Mrkvička T, Muška M, Kubečka J (2014). "Two Step Estimation for Neyman-Scott Point Process with Inhomogeneous Cluster Centers." *Statistics and Computing*, **24**(1), 91–100. doi:10.1007/s11222-012-9355-3.

Mrkvička T, Myllymäki M (2023). "False Discovery Rate Envelopes." *Statistics and Computing*, **33**, 109. doi:10.1007/s11222-023-10275-7.

Myllymäki M, Mrkvička T (2023). "**GET**: Global Envelopes in R." arXiv:1911.06583 [stat.ME]. doi:10.48550/arXiv.1911.06583.

Affiliation:

Mari Myllymäki
Natural Resources Institute Finland (Luke)
Latokartanonkaari 9
FI-00790 Helsinki, Finland
E-mail: mari.myllymaki@luke.fi
URL: https://www.luke.fi/en/experts/mari-myllymaki/
and
Tomáš Mrkvička
Dpt. of Applied Mathematics and Informatics
Faculty of Economics
University of South Bohemia,
Studentská 13
37005 České Budějovice, Czech Republic
E-mail: mrkvicka.toma@gmail.com
URL: http://home.ef.jcu.cz/~mrkvicka/