

Analysis of a Point Pattern in R

Facundo Muñoz

10/06/2014

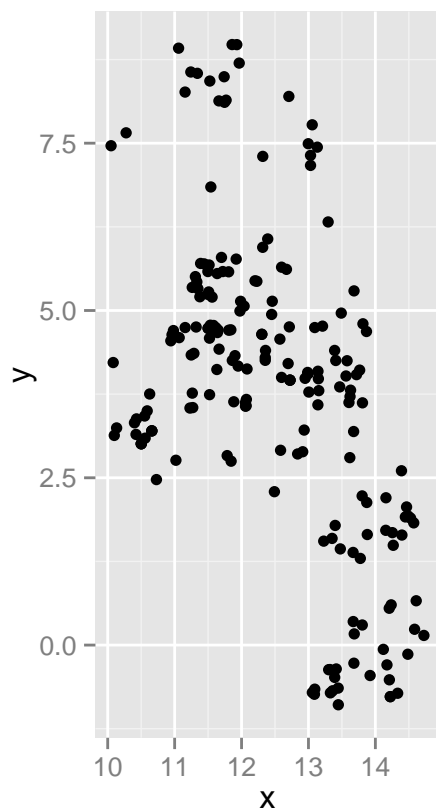
Sample script to show how to analyse a Point Pattern in R. We model the dataset with an inhomogeneous Poisson Process, estimate its intensity function, and compute its gradient rate.

Read coordinates from csv. Units are meters? Rescale to km.

```
coord <- read.csv('../data/faunacoordinates.csv',  
                  header = FALSE,  
                  col.names = c('x', 'y'))  
coord <- coord/1e3
```

Take a look at the data.

```
ggplot(coord, aes(x, y)) +  
  geom_point() +  
  coord_fixed()
```



Create a `spatstat`'s Point Pattern object from the coordinates.

```
# Define region as a rectangle containing all the points  
# with limits extended by 4%  
extend.range <- function(x, pct) {
```

```

range <- range(x)
range + c(-1, 1) * pct * diff(range) / 100
}
pct = 4
lims <- sapply(coord, extend.range, pct)
dat.ppp <- as.ppp(coord, as.owin(c(lims)))

```

```

## Warning in ppp(X[, 1], X[, 2], window = win, marks = marx, check = check):
## data contain duplicated points

```

```

mean.intensity <- with(dat.ppp, n/diff(window$xrange)/diff(window$yrange))

```

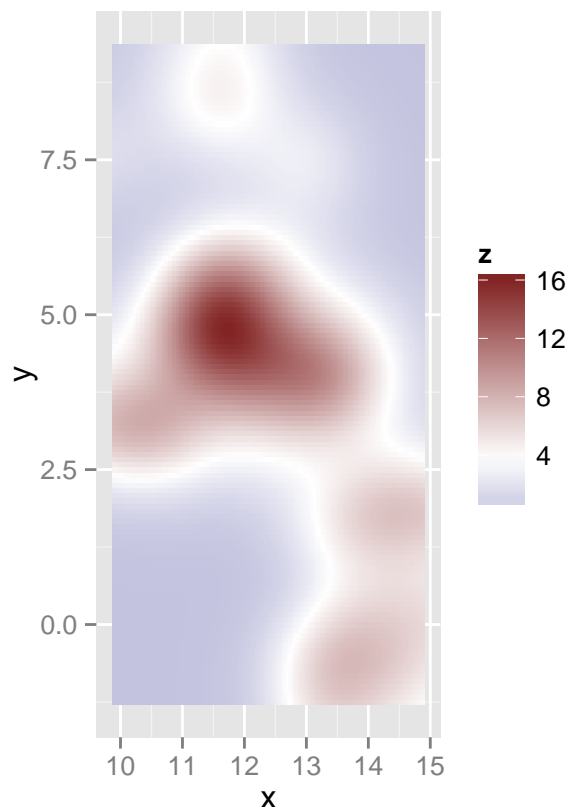
The *mean intensity* is the average density of points over the region, and takes the value 3.6989873 in this case.

Compute a Kernel Density Estimate of the intensity function. See `?spatstat::density.ppp` for details

```

intensity <- density(dat.ppp)
# plot(intensity, main = 'Estimate of intensity')
# Nicer:
ggplot(with(intensity, cbind(expand.grid(x = xcol, y = yrow),
                               z = c(t(v))))),
       aes(x, y, fill = z)) +
  geom_raster() +
  scale_fill_gradient2(low = "#3A3A98FF",
                      high = "#832424FF",
                      midpoint = mean.intensity) +
  coord_fixed()

```



Compute the slope of the intensity surface. As far as I understand, this is what Maximiano Castillejo calls *spatial gradient*. Here it is important to make sure that *x* and *y* are *projected* coordinates, and not lat/lon.

```
# transform the image as a RasterLayer object
# just use the EPSG code to state that coordinates are planar
intens.rst <- raster(intensity, crs = CRS("+init=epsg:3732"))
slope.rst <- terrain(intens.rst)
# plot(slope.rst, main = 'Slope of intensity')
# Nicer:
ggplot(data.frame(coordinates(slope.rst),
                      z = slope.rst@data@values),
        aes(x, y, fill = z)) +
  geom_raster() +
  coord_fixed()
```

