Week 3 Assignment

Auriel M.V. Fournier
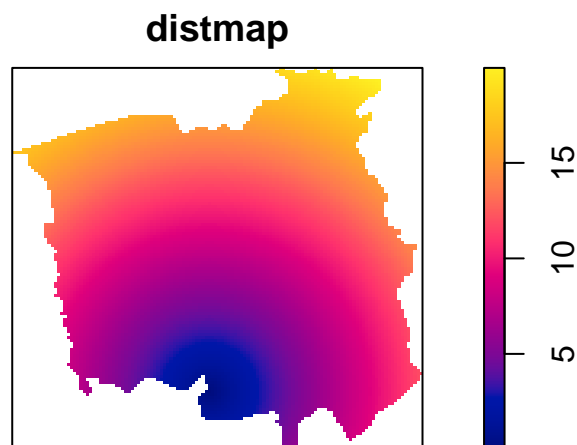
```
library(spatstat)
```

```
##
## spatstat 1.40-0        (nickname: 'Do The Maths')
## For an introduction to spatstat, type 'beginner'
```

```
data(chorley)

ch <- unique.ppp(chorley)
ch_lung <- ch[ch$marks=="lung",]
ch_lar <- ch[ch$marks=="larynx",]
this.window <- ch$window
incin <- ppp(chorley.extra$incin$x, chorley.extra$incin$y, window=this.window)
distmap <- distmap(incin)

plot(distmap)
```
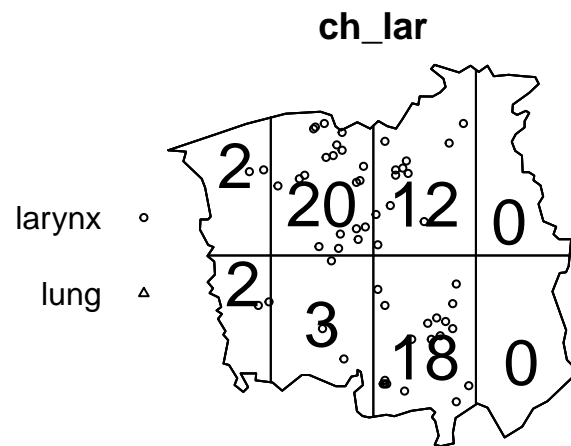

distmap

```r
lambda <- summary(ch)$intensity

q <- quadratcount(ch, nx=4,ny=2)
q_lung <- quadratcount(ch_lung, nx=4,ny=2)
q_lar <- quadratcount(ch_lar, nx=4,ny=2)

plot(ch_lar, cex=0.5)
plot(q_lar, add=T, cex=2)
```
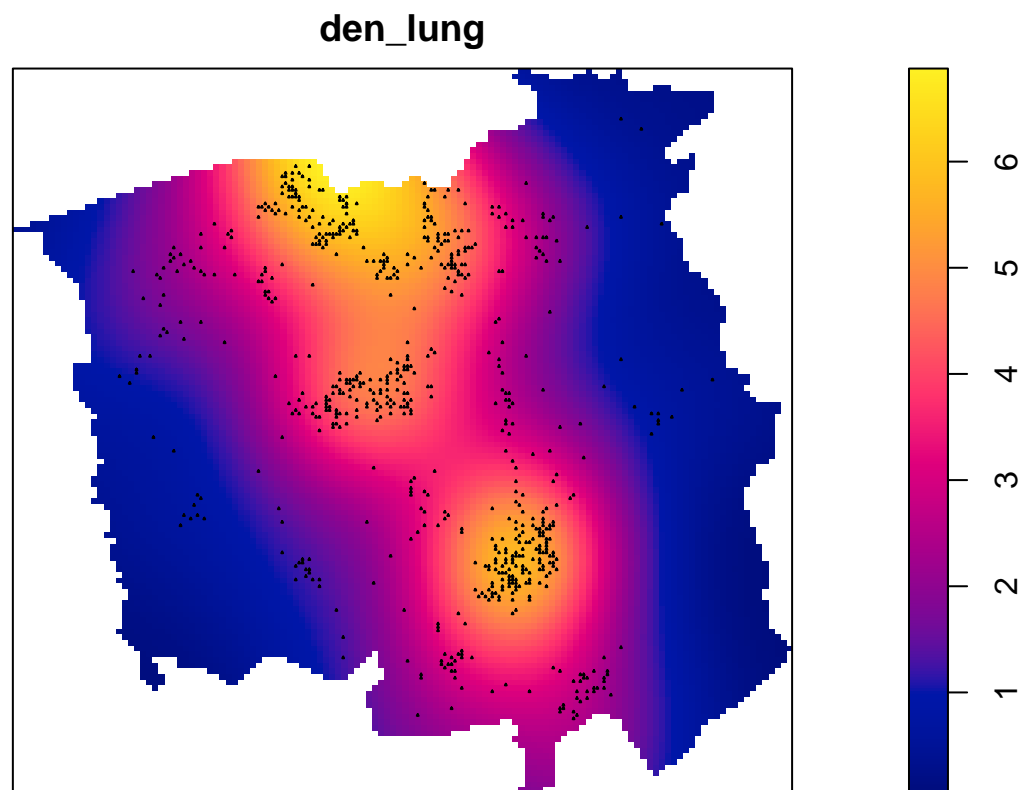

ch_lar

```r
#kde to create a surface predicting population over the
#entire surface using lung cancer occurances as a subsittute for population
den_lung <- density.ppp(ch_lung, sigma=2, kernel='gaussian')

plot(den_lung)
plot(ch_lung, add=T, cex=.1)
```
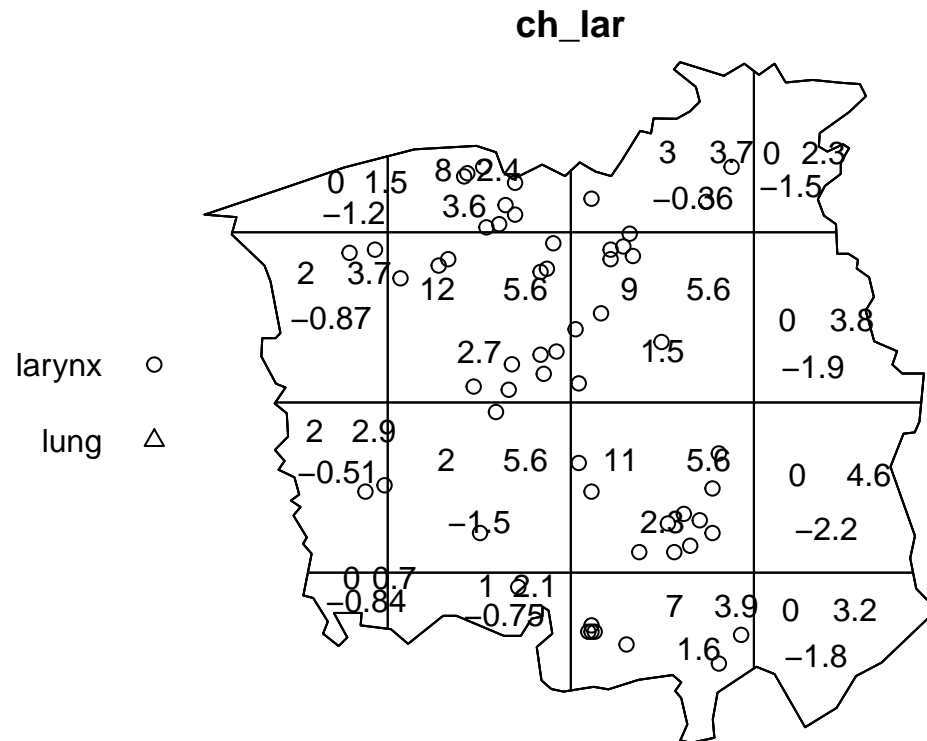
den_lung

```
csr.test <- quadrat.test(ch_lar, nx=4, ny=4, method="Chisq")
```

```
## Warning: Some expected counts are small; chi^2 approximation may be
## inaccurate
```

```
plot(ch_lar, color='blue')
plot(csr.test, add=T, color="red")
```

ch_lar

larynx   ○

lung   △

```
csr.test
```

```
##
##  Chi-squared test of CSR using quadrat counts
##  Pearson X2 statistic
##
## data:  ch_lar
## X2 = 50.2628, df = 15, p-value = 2.181e-05
## alternative hypothesis: two.sided
##
## Quadrats: 16 tiles (irregular windows)
```

So the p-value is really small, meaning that they are not Completely Spatially Random.

```
null <- ppm(ch_lar)
dist <- ppm(ch_lar, ~distmap, covariates=distmap)
pop <- ppm(ch_lar, ~den_lung, covariates=den_lung)
pop_dist <- ppm(ch_lar, ~den_lung+distmap, covariates =c(den_lung, distmap))
pop.dist <- ppm(ch_lar, ~den_lung*distmap, covariates=c(den_lung, distmap))

aic <- matrix(ncol=2, nrow=5)
aic[,1] <- c("null","dist","pop","pop_dist","pop.dist")
aic[,2] <- c(AIC(null),AIC(dist),AIC(pop),AIC(pop_dist),AIC(pop.dist))

aic
```

```
##      [,1]       [,2]
## [1,] "null"     "389.960404836208"
## [2,] "dist"     "390.250675338287"
## [3,] "pop"      "329.189321038627"
## [4,] "pop_dist" "328.255917047726"
## [5,] "pop.dist" "329.581398989165"
```

```
aic[aic[,2]==min(aic[,2]),]
```

```
## [1] "pop_dist"        "328.255917047726"
```

Based on the AIC of these models the model with 328.255 is the lowest and is for the pop_dist model.

First lets just look at the model output for pop_dist

```
pop_dist
```

```
## Nonstationary multitype Poisson process
##
## Possible marks: 'larynx' and 'lung'
##
## Log intensity:  ~den_lung + distmap
##
## Fitted trend coefficients:
## (Intercept)    den_lung     distmap
## -3.88937374  0.63771937 -0.05733376
##
##               Estimate       S.E.    CI95.lo      CI95.hi Ztest      Zval
## (Intercept) -3.88937374 0.42485714 -4.7220784 -3.056669048   *** -9.154545
## den_lung     0.63771937 0.08669755  0.4677953  0.807643451   ***  7.355679
## distmap     -0.05733376 0.03346583 -0.1229256  0.008258061       -1.713203
```

So there is a positive response to den_lung (our proxy for population) so it looks like there is more occurances of larynx where there are more people.

This model includes the distmap covariate, but it is not significant.

If we look at the model with only population we see that it has a slightly higher AIC, but it's with 2 Delta AIC, so that is another plausible model. It has a similar positive value for it's parameter estimate, a slightly lower value, but similar and still significant.

From this result I would say that larynx cancer rates are not related to distance from the incinerator, but are occuring at a certain rate within the population, perhaps related to some other factor that we do not have the data to understand right now.

I wasn't able to get the predict() function to produce predictions, or it did, but it didnt produce standard errors.

```r
pop_den_predict <- predict(pop_dist, type="intensity", se=T, window=this.window)
names(pop_den_predict)
```

```
## [1] "larynx" "lung"
```

```r
plot(pop_den_predict$larynx)
```

pop_den_predict$larynx