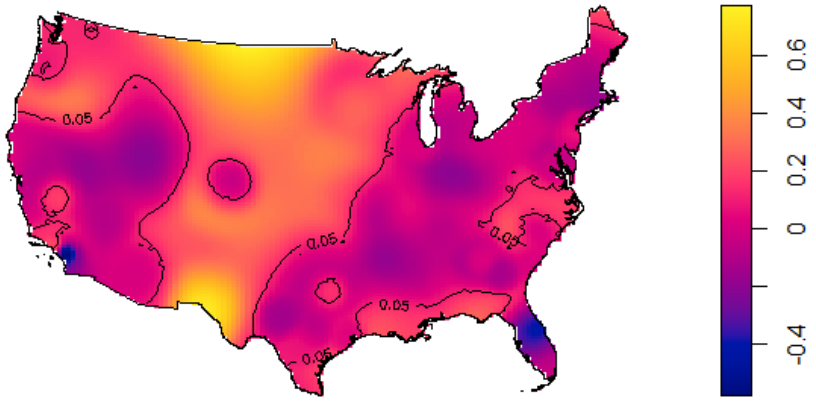
W7 Exercise: Kernel Density Estimation

OBJECTIVE. Use the Sparr package in R to compute the Spatial Relative Risk/Density Ratio Function. Total: 40pts.



In this exercise, you are using the Sparr (Spatial and Spatiotemporal Relative Risk) package in R by Tilman M. Davies to compute and visualize kernel density estimates for case/control data (Davies et al. 2011). Many geographic phenomena like tweets, disease or crime emerge out of a population and manifest themselves as point locations (tweet location, residential locations of patients, street address of crimes). We expect to observe more points (cases) in heavily populated areas, which is why we need to account for the underlying population (controls) in kernel density estimation. If we omit to account for the underlying population, the pattern of our maps will always be the same: high density of the phenomenon of interest in locations where the underlying population is dense. Therefore, the Spatial Relative Risk/Density Ratio Function calculates the ratio of case and control densities, which allows us to draw maps that show areas where case density is unexpectedly high (Shi 2010).

In this tutorial, we are using a dataset of geolocated tweets that have been classified into *adverse* (cases) and *non-adverse* (controls). This outcome relates to sentiment expressed in the tweet body and has been classified using a dictionary-based approach. Details on the sentiment classification are not relevant to this exercise but can be obtained by asking Alexander Hohl.

Let's get started!

1. Download and unzip the W9_Exercise_data.zip
2. Create an R script and start typing code

Import the necessary libraries (install them, if necessary)

```
library(sparr)
library(readxl)
library(dplyr)
library(sf)
```

Read the tweets dataset, extract the relevant columns and convert to a sf object using the coordinates in the *x* and *y* columns:

```
tweets <- read_xlsx("data/tweets/Tweet_2019_12_6424.xlsx") %>%
      subset(., select=c("x", "y", "adverse")) %>%
      st_as_sf(., coords = c("x","y"))
```

The coordinates in the tweets dataset are in WGS84, so set the CRS of the tweets object accordingly. Conveniently, each CRS has an identifying number, in this case it is 4326:

```
st_crs(tweets) = 4326
```

Transform the coordinates to U.S Atlas Equal Area (crs=2163):

```
tweets_proj <- st_transform(tweets, crs = 2163) %>%
      st_coordinates(.) %>%
      cbind(.,tweets$adverse) %>%
      as.data.frame(.)
colnames(tweets_proj) <- c("x", "y", "adv")
```

Read the contiguous United States boundary shapefile and transform to U.S Atlas Equal Area:

```
conus <- st_read("data/tweets/CONUS_diss.shp") %>%
      st_transform(., crs=2163)
```

Educate yourself on the fantastic functions that the sparr package has to offer. Browse through the reference page, and pay special attention to the Spatial relative risk/density ratio function, which we are using here. The function takes a point pattern (ppp) object with dichotomous factor-valued marks, which distinguish cases and controls. Therefore, we need to wrangle our data into this format first:

```
tw_ppp <- ppp(tweets_proj$x, tweets_proj$y, marks = as.factor(tweets_proj$adv),
      window = as.owin(conus))
```

As you can see, we took z and y coordinates, as well as the marks from the `tweets_proj` object, and supplied the study window from the `conus` object. We use the study window merely to provide context for plotting, but it does have relevance for point pattern analysis methods that are affected by boundary conditions (i.e. Ripley's K function).

Next, we calculate the global bandwidth according to the oversmoothing principle (Terrell 1990):

```
h0 <- OS(tw_ppp,nstar="geometric")
```

We will need the global bandwidth later when calculating the Spatial relative risk/density ratio function. Next, we split the point pattern object into separate objects for cases and controls:

```
twAdv <- tw_ppp[tw_ppp$marks == "1"] #adverse tweets
twNor <- tw_ppp[tw_ppp$marks == "0"] #normal tweets
```

It's time to compute the kernel density estimates using the Spatial relative risk/density ratio function. There are multiple different ways to do it, especially regarding bandwidth selection. We will explore four different bandwidth regimes: First, we use a fixed symmetric bandwidth, meaning the bandwidth is the same across the entire study area, and it is the same for both, that case- and control densities. Here, we are using the global bandwidth which we calculated in the previous step:

```
tweet_dens1 <- risk(f=twAdv, g=twNor, h0=h0, tolerate=TRUE, doplot=TRUE)
```

Note that the resulting map contains tolerance contours, which we implemented by setting the `tolerate` parameter to `TRUE`.

Next, we use asymmetric adaptive bandwidths. This means that the bandwidths vary, as they adapt to case and control densities independently. We implement this behavior by specifying a common global bandwidth (`h0` parameter), but separate pilot bandwidths for case- and control densities (`hp` parameter), which are again chosen according to the oversmoothing principle (Terell 1990):

```
tweet_dens2 <- risk(f=twAdv, g=twNor, h0=h0, adapt=TRUE, hp=OS(tw_ppp)/2,
      tolerate=TRUE, davies.baddeley=0.05, doplot=TRUE)
```

Note that the asymmetric adaptive bandwidths are not recommended (Davies, Jones & Hazelton 2016). This bandwidth selection regime can cause artefacts in the resulting density maps in form of halos of increased risk. Can you spot the halos in the map?

Further, we use symmetric (pooled) adaptive bandwidths: The bandwidths adapt to the underlying point data, but are always the same for case and control densities. We achieve this behavior by specifying a common global bandwidth (as calculated in the previous step), and common pilot bandwidths. In addition, we need to set the value for the `pilot.symmetry` parameter to `"pooled"`:

```
tweet_dens3 <- risk(f=twAdv, g=twNor, h0=h0, adapt=TRUE, tolerate=TRUE,
      hp=OS(tw_ppp)/2, pilot.symmetry="pooled", davies.baddeley=0.05,
      doplot=TRUE)
```

Lastly, we use symmetric bandwidths that adapt to the control density. In this example, we illustrate how the `risk()` function can take `bivariate.density()` objects directly, instead of taking `ppp` objects to calculate the Spatial relative risk/density ratio function:

```
f <- bivariate.density(twAdv,h0=h0,hp=2,adapt=TRUE,pilot.density=twNor,
      verbose=FALSE)
g <- bivariate.density(twNor,h0=h0,hp=2,adapt=TRUE,pilot.density=twNor,
      verbose=FALSE)
tweet_dens4 <- risk(f=f,g=g,tolerate=TRUE,log=TRUE,doplot=TRUE)
```

You are now able to perform kernel density estimation for case/control data using different bandwidth selection regimes (symmetric vs. asymmetric, fixed vs. adaptive). To obtain the 40 points of this exercise, you need to implement the Spatial relative risk/density ratio function using your own data. Your task is to find a case/control dataset (or use one of the datasets provided on CANVAS, tweets excluded) and draw four maps using different bandwidth selection regimes (corresponding to the examples in this tutorial).

You should submit:

- An assignment report that contains the four maps, as well as one paragraph each describing the pattern, and another paragraph where you argue which of the 4 regimes is the best for your data.
- A zip file containing your code and data

References:

- Davies, T. M., Hazelton, M. L., & Marshall, J. C. (2011). Sparr: analyzing spatial relative risk using fixed and adaptive kernel density estimation in R. Journal of Statistical Software, 39(i01).
- Shi, X. (2010). Selection of bandwidth type and adjustment side in kernel density estimation over inhomogeneous backgrounds. International Journal of Geographical Information Science, 24(5), 643-660.
- Terrell, G.R. (1990), The maximal smoothing principle in density estimation, Journal of the American Statistical Association, 85, 470-477.
- Davies, T. M., Jones, K., & Hazelton, M. L. (2016). Symmetric adaptive smoothing regimens for estimation of the spatial relative risk function. Computational Statistics & Data Analysis, 101, 12-28.