Exercise 1 – Moran's I and Gearys's C

Consider the leukaemia data from Exercise sheet 1, Exercise 2. Additionally, several covariates were recorded for each population centroid, e.g. the sex ratio of the residents or the percentage of residents having a certain age. Here it is of interest, if the average age is spatially correlated within the study region.

- Load the `R` packages `fields` and `spdep` and read in the dataset `nycovariates.txt`. Look at its variables using `names()`; one is called `averageage`. Eastings and northings are stored in the variables `xcoord` and `ycoord`.
- Chose one strategies to define the weight matrix you need for a calculation of Moran's I and Geary's C (see the lecture notes). The functions `tri2nb()`, `knearneigh()` and `nb2listw()` can be helpful.
- Perform a test for spatial autocorrelation using Moran's I - you can perform a test assuming normality (`moran.test()`) or a Monte Carlo test (`moran.mc()`). Compare the results of the different tests.
- Repeat the testing for Geary's C and interpret the results. The relation between Moran's I and Geary's C is complex, but it appears that C is sensitive to particularly large differences between values in adjacent regions, while extreme response values have a greater impact on I.

Exercise 2 – Test CSR of a point pattern using a χ^2-test

In the years 1974-1983 locations of lung and larynx cancer cases in Chorley-Ribble, Lancashire, were recorded. It is of interest, if the observed lung cancer cases form a random pattern.

- Load the `R` libraries `spatstat` and `splancs` and load the dataset `southlancs`. The first two columns contain the grid eastings and northings of the cancer cases. The third column contains an indicator, if a lung or a larynx cancer case was observed (lung=0, larynx=1). Extract all lung cancer cases from the dataset and plot them. Create a `ppp()` object from these data. The dimensions of the study area (`southlancs.bdy`) have to be specified using `xrange` and `yrange`.
- Perform a χ^2-test to test the complete spatial randomness (CSR) of the lung cancer cases: The study region is partitioned into n squares and it is tested by the χ^2-test, if the observed number of cases in a square differs from the expected number of cases under CSR. To perform this test use the function `quadrat.test()`. Interpret its results; what is problematic?

Exercise 3 – Stone's Test for the IBM Endicott waste site

Consider the leukaemia data from Exercise sheet 1, Exercise 2. Your hypothesis is that cases cluster around the IBM Endicott waste site, which is located at $(-4.47, -67.65)$.

- Load the package `DCluster` and read in the dataset `newyork2.txt`.
- To be used by the library `DCluster` the data have to be stored in a `data.frame()` including the variables `Observed, Expected, Population, x, y`. The number of expected cases per region is not given, but can be calculated by internal standardization, multiplying the population in each district by the ratio

$$\sum_i (\text{cases per district } i) / \sum_i (\text{population per district } i).$$

- Find the population centroid which is closest to the IBM Endicott waste site using function `knearneigh()` with option `k=1`.
- Perform Stone's test for the IBM Endicott waste site with the closest population centroid as reference point using function `stone.test()` and interpret the result.

Exercise 4 – Detection of clusters using the scan statistic

Besides the method of Besag and Newell there is another popular approach to detect clusters in a study area: the scan statistic. It has the advantage that the clusters we are looking for don't have to have a prespecified size. The study region is scanned by circular regions around the population centroids with varying radius - starting with radius 0 up to a size of the circle which includes a prespecified fraction of the total population (`fractpop`). Consider again the leukaemia data from Exercise sheet 1, Exercise 2.

- Look at the lecture notes showing the example for the method of Besag and Newell and use the same `R` code for detection of clusters using the scan statistic, just by exchanging `bn.iscluster` by `kn.iscluster`. Use `fractpop=0.2` and only `R=50` simulation repetitions. As input data you can use the `data.frame` from Exercise 3.
- In order to find the most important clusters, the clusters can be sorted by the value of the scan test `statistic`. You can use the function `order()` to obtain an order of your clusters with regard to the test statistic.
- Plot the locations of the population centroids and the 10 most important cluster centres. Add the locations of all waste sites to your plot; they are stored in the dataset `wastesites.txt`. Interpret your results. Compare the results of the scan statistic with the result of the method by Besag and Newell!

Exercise 5 – (*) Implementing a Monte Carlo Test

It should be tested, if larynx cancer cases observed in Chorley-Ribble, Lancashire, form a random point pattern. As a test statistic the smallest distance between two points of the dataset was chosen. Since the distribution of the smallest distance between two points of a point pattern is not known, a Monte Carlo test has to be performed.

- Load the dataset `southlancs` and extract the larynx cancer cases.
- Simulate a random point pattern within the study region `southlancs.bdy` using the function `csr()` and plot it.
- Simulate 999 random point patterns, compute the minimum distance between all pairs of points of each simulated pattern and store it in a vector. The distance between two points of a pattern can be calculated using the function `rdist()`.
- Compute the minimum distance for the observed larynx cancer pattern.
- Order the vector containing the 1000 distance minima using `order()` and compute the rank of the observed point pattern.
- Compute the Monte Carlo p-value using the formula from the lecture notes. Is the observed larynx cancer pattern a realisation of a random process?