Spatial Statistics

Computer lab – Session 1

Madlene Nussbaum Andreas Papritz (ehem. ETHZ)

07 Oct 2024

Table of contents

1 Install required software **1**

2 Spatial data in R – classes and plotting **2**
2.1 Spatial data types 2
2.2 Package `sf` 2
2.3 Package `terra` 3
2.4 Creating maps 3
2.5 Package `sp` and more maps 4

3 First steps of geostatistical analysis **5**
3.1 Exploratory analysis 5
3.2 Fitting a trend model 5
3.3 Exploring and modelling auto-correlation 6

For the following exercises, use the R code of the lecture slides on the introduction to spatial data and the analysis of the Wolfcamp aquifer dataset as template.

1 Install required software

```
install.packages("spDataLarge", repos = "https://geocompr.r-universe.dev")

l.p <- c("Boruta", "caret", "data.table", "geoGAM", "georob", "ggplot2",
```

```
         "gstat", "mapview", "ranger","rgl", "sf", "sp", "terra", "tidyterra")
install.packages(l.p)
```

2 Spatial data in R – classes and plotting

In this section you will get familiar with the main R packages to handle spatial data, how to inspect the content and how to plot spatial data.

2.1 Spatial data types

The main spatial data types used for geostatistical tasks are point vector geometries and raster data.

> Task 1
>
> Familiarize yourself with spatial data types. Read through Pebesma and Bivand, 2023. Section 1, Getting started.

Coordinate reference systems (CRS) are often referred to by a identifying number (EPSG). An overview of worldwide CRS can be found here https://epsg.io.

2.2 Package `sf`

R package `sf` handles geographical vector data. Get familiar with `sf` objects by reading Pebesma and Bivand, 2023. Section 7.1 (up to and including *Subsetting*).

> Task 1
>
> Load the landslide dataset `lsl` from package `spDataLarge`. Create a spatial `sf` object using `st_as_sf()` and assign the correct CRS. See information on the dataset's help page.

> Task 2
>
> Extract the spatial coordinates as vectors (this is often needed to do non-spatial plotting or handing over to functions that cannot deal with `sf` objects).

Task 3

Create a simple plot of the landslide points (`true`: landslide initiation points, `false`: unaffected).

2.3 Package `terra`

R package `terra` handles geographical raster data. An alternative is the `stars` package which is yet less widespread. The predecessor of `terra` is the no longer supported `raster` package.

Read in the elevation data from `spDataLarge`.

```
library(terra)
ta <- rast(system.file("raster/ta.tif", package = "spDataLarge"))
```

Task 1

Inspect the content of the elevation raster. Access e.g. the first 10 pixel values of one layer by using `values()`.
Optional: Set all values > 2000 m in the elevation raster to `NA`.

Task 2

Create a simple plot of the slope (first band).

2.4 Creating maps

Being able to display geographical data prevents erroneous analysis. Besides classical x-y-plotting, R offers a large variety of display options for spatial data. Above you already created maps using `plot` function. Besides, we will use `ggplot2` and `mapview` packages.

The R package `tmap` also offers many possibilities, if interested check out Lovelace et al. 2024, Chapter Making Maps with R.

Task 1

Using the elevation and landslide data loaded above: create a map showing the slope and overlay the landslide observations points (`points()`).

For mapping with `ggplot2` see option in chapter Pebesma and Bivand, 2023. Maps with ggplot2. For `terra` objects see here tidyterra. For extended examples to map with `ggplot2` see Wickham, et al., 2024.

> Task 2
>
> Plot the elevation and landslide data using `ggplot2`. To display terra objects with `ggplot2` use `geom_spatraster()` from the `tidyterra` package. `sf` objects can be plotted with `geom_sf()`.

2.5 Package `sp` and more maps

The R package `sp` has been replaced by the easier to use and more powerful `sf` package (published by the same authors). Geostatistical R packages, however, still depend on spatial `sp` objects and sometimes require `sp` objects as input.

How to transform `sf` to `sp` and vice versa see Pebesma and Bivand, 2023. Appendix A — Older R Spatial Packages.

> Task 1
>
> Load the dataset `meuse` from package `sp`. Create a spatial object and assign the correct CRS. See `example` on the dataset's help page.

> Task 2
>
> Plot the `meuse` observation locations using `plot`. Plot the topsoil zinc concentrations with `spplot`.

> Task 3
>
> Create an interactive map displaying the zinc concentration using the `mapview` package. Change the view to aerial imagery.

> Task 4
>
> Similarly, load `meuse.grid`. Create a `sp` raster object (`SpatialPixelsDataFrame`). Explore the content by printing and plotting.

3 First steps of geostatistical analysis

3.1 Exploratory analysis

We are now going to have a closer look into the elevation data contained in the package `georob`:

```
library(georob)
data(elevation)
```

> Task 1
>
> Plot `height` against the `x-` and `y-`coordinates. Consider adding a smooth LOESS curve to the scatterplots.
> Create a "bubble plot" to explore the spatial distribution of the response variable. For the bubble plot, choose the size of the symbols such that their area depends linearly on `height`.

> Task 2
>
> Explore the spatial distributions of `height` further by the dynamic graphics of the package `rgl`.

> Task 3
>
> Does `height` show some large-scale trend? If yes, how would you build a regression model to model this trend?

3.2 Fitting a trend model

> Task 1
>
> Fit the trend model that you found in the previous problem now by ordinary least squares.

> Task 2
>
> Use customary residual diagnostics to see if the model can be improved.

Task 3

Create a bubble plot of the residuals to see if the residuals are show spatial auto-correlation.

3.3 Exploring and modelling auto-correlation

Task 1

Using the function `hscat()` of the package `gstat`, create lag-scatter plots of the regression residuals. To use `hscat()` you have to convert the dataframe `elevation` to a `SpatialPointsDataFrame` (package `sp`).

Task 2

Compute the sample variogram of the residuals and estimate the size of the nugget, total sill and range from the plot of the sample variogram.

Task 3

Fit a "spherical" variogram function to the sample variogram. Use your guess estimates of the variogram parameters as initial values for model fitting.
Do the fitted parameter values differ much from your guesses?