C10-2 Preprocessing of spatial data sets with R

Preprocessing of spatial data sets wiht R

The following examples illustrate the preprocessing of spatial data sets using the sp and raster package.

As always, we start with importing the packages and setting the working directory.

```
library(raster)
```

```
## Loading required package: sp
```

```
library(sp)
library(rgdal)  # Necessary to read vector files when reading with sp
```

```
## rgdal: version: 0.8-16, (SVN revision 498)
## Geospatial Data Abstraction Library extensions to R successfully loaded
## Loaded GDAL runtime: GDAL 1.11.0, released 2014/04/16
## Path to GDAL shared files: C:/Users/tnauss/Documents/R/win-
library/3.0/rgdal/gdal
## GDAL does not use iconv for recoding strings.
## Loaded PROJ.4 runtime: Rel. 4.8.0, 6 March 2012, [PJ_VERSION: 480]
## Path to PROJ.4 shared files: C:/Users/tnauss/Documents/R/win-
library/3.0/rgdal/proj
```

```
setwd("D:/active/moc/dm/programs/r-dm-ln-12")
```

Preprocessing of xyz data (sp package)

XYZ data sets are just csv files. Hence, one can easily read them.

```
lidar <- read.table("lidar_example.txt", header=TRUE, sep=",")
str(lidar)
```

```
## 'data.frame':    125968 obs. of  8 variables:
##  $ X                  : num  481167 481216 481307 481966 481691 ...
##  $ Y                  : num  5646000 5646000 5646000 5646000 5646000 ...
##  $ Z                  : num  263 285 291 358 339 ...
##  $ Classification     : int  13 13 13 13 13 13 2 2 2 2 ...
##  $ Classification.Name: Factor w/ 2 levels "Ground","Reserved for ASPRS
Definition": 2 2 2 2 2 2 1 1 1 1 ...
```

```
##  $ Number.of.Returns  : int  1 1 2 2 1 1 1 1 1 1 ...
##  $ Return.Number      : int  1 1 1 1 1 1 1 1 1 1 ...
##  $ Intensity          : int  42 22 43 14 12 17 57 33 91 84 ...
```

As one can see in the structural information, no spatial reference is supplied with the data set. This can also be checked using the projection function.

```
projection(lidar)
```

```
## [1] NA
```

In order to visualize the data in a map-related format, a spatial reference system must be supplied. In addition, one has to define the columns of the data frame which provide the east and north values.

Note: Since the variable lidar still represents a data frame and not an sp package type of *spatial data frame*, the function `coordinates` has to be executed first because it returns exactly this kind of spatial data frame. If `projection` is exectued first, it will not find the appropriate data structure for writing the spatial reference information.

```
coordinates(lidar) <- ~X+Y
projection(lidar) <- "+proj=utm +zone=32 +ellps=GRS80 +units=m +north"
str(lidar)
```

```
## Formal class 'SpatialPointsDataFrame' [package "sp"] with 5 slots
##   ..@ data       :'data.frame':  125968 obs. of  6 variables:
##   .. ..$ Z                  : num [1:125968] 263 285 291 358 339 ...
##   .. ..$ Classification     : int [1:125968] 13 13 13 13 13 13 2 2 2 2
...
##   .. ..$ Classification.Name: Factor w/ 2 levels "Ground","Reserved for
ASPRS Definition": 2 2 2 2 2 2 1 1 1 1 ...
##   .. ..$ Number.of.Returns  : int [1:125968] 1 1 2 2 1 1 1 1 1 1 ...
##   .. ..$ Return.Number      : int [1:125968] 1 1 1 1 1 1 1 1 1 1 ...
##   .. ..$ Intensity          : int [1:125968] 42 22 43 14 12 17 57 33 91
84 ...
##   ..@ coords.nrs : int [1:2] 1 2
##   ..@ coords     : num [1:125968, 1:2] 481167 481216 481307 481966 481691
...
##   .. ..- attr(*, "dimnames")=List of 2
##   .. .. ..$ : NULL
##   .. .. ..$ : chr [1:2] "X" "Y"
##   ..@ bbox       : num [1:2, 1:2] 481000 5646000 482000 5647000
##   .. ..- attr(*, "dimnames")=List of 2
##   .. .. ..$ : chr [1:2] "X" "Y"
##   .. .. ..$ : chr [1:2] "min" "max"
##   ..@ proj4string:Formal class 'CRS' [package "sp"] with 1 slots
##   .. .. ..@ projargs: chr "+proj=utm +zone=32 +ellps=GRS80 +units=m"
```

The summary of the data set now additionally contains the spatial reference.

```
summary(lidar)
```

```
## Object of class SpatialPointsDataFrame
## Coordinates:
##       min     max
## X  481000  482000
## Y 5646000 5647000
## Is projected: TRUE
## proj4string :
## [+proj=utm +zone=32 +ellps=GRS80 +units=m]
## Number of points: 125968
## Data attributes:
##        Z       Classification                    Classification.Name
##  Min.   :233   Min.   : 2.00   Ground                        :45666
##  1st Qu.:282   1st Qu.: 2.00   Reserved for ASPRS Definition:80302
##  Median :303   Median :13.00
##  Mean   :302   Mean   : 9.01
##  3rd Qu.:325   3rd Qu.:13.00
##  Max.   :385   Max.   :13.00
##  Number.of.Returns Return.Number    Intensity
##  Min.   :1.00      Min.   :1.00   Min.   : 10.0
##  1st Qu.:1.00      1st Qu.:1.00   1st Qu.: 15.0
##  Median :1.00      Median :1.00   Median : 22.0
##  Mean   :1.51      Mean   :1.25   Mean   : 33.7
##  3rd Qu.:2.00      3rd Qu.:1.00   3rd Qu.: 38.0
##  Max.   :5.00      Max.   :5.00   Max.   :223.0
```

Preprocessing of a polygon vector (sp package)

In contrast to the xyz data set, polygon vector files in the ESRI shape format generally contain a file ending on prj which provides the projection information.

For reading the data set, the readOGR function of the sp package is used.

```
polygon <- readOGR("polygon.shp", layer = "polygon")
```

```
## OGR data source with driver: ESRI Shapefile
## Source: "polygon.shp", layer: "polygon"
## with 16 features and 3 fields
## Feature type: wkbPolygon with 2 dimensions
```

```
projection(polygon)
```

```
## [1] "+proj=utm +zone=32 +ellps=GRS80 +units=m +no_defs"
```

Preprocessinf of raster files (raster package)

As for other GIS data formats, a Geo TIFF generally supplies meta information on its reference system, too. This is also the case for that landast scene.

```
intensity <- raster("intensity_1rt1.tif")
projection(intensity)
```

```
## [1] "+proj=utm +zone=32 +ellps=GRS80 +units=m +no_defs"
```

But what if no reference system is supplied? No problem. Just assign it as for any other data set. Of course, one has to delete the information first if one uses this example data set.

```
projection(intensity) <- NA
projection(intensity) <- "+proj=utm +zone=32 +ellps=GRS80 +units=m +no_defs"
```