Analysis of spatial data in R

IMPRS R course: Advanced modules

Spatial data in R

Matthias Forkel

mforkel@bgc-jena.mpg.de

Jena, 24.04.2015

Presentation, example scripts, data and materials at:

ftp://ftp.bgc-jena.mpg.de/pub/outgoing/mforkel/Rcourse/

Agenda

Max Planck Institute for Biogeochemistry

1 Introduction to spatial data in R
1.2 Packages for spatial data
1.3 Classes for vector and raster datasets
1.4 Reading and writing of spatial data

2 Mapping
2.1 Plotting functions for spatial data
2.2 Graphical representation of spatial information
2.3 Coordinate systems and reprojection

3 Spatial data analysis
3.1 Methods for vector data
3.2 Methods for raster data

4 Exchange between R and other spatial software
4.1 Import of GoogleMaps and OpenStreetMap data in R
4.2 Export data to GoogleEarth
4.3 Linking R with GIS software

optional

Introduction to spatial data in R

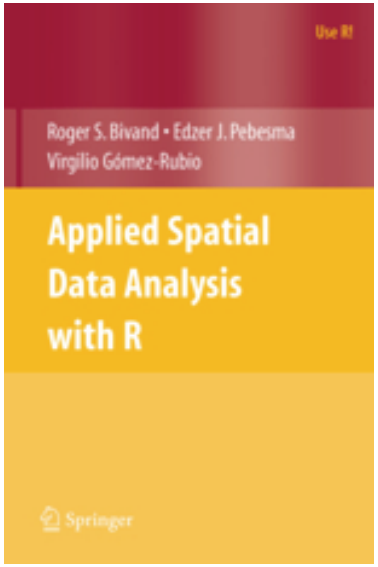
as GIS? – Why and how is this possible?

Max Planck Institute for Biogeochemistry

- **Geostatistics as a core feature of R:**
  - Spatial autocorrelation, spatial interpolation, geostatistics (package „spatial“ is a core part of S/R for a long time)
  - visualization capabilities of R: mapping
  - many packages for geostatistical analyzes (e.g. gstat)

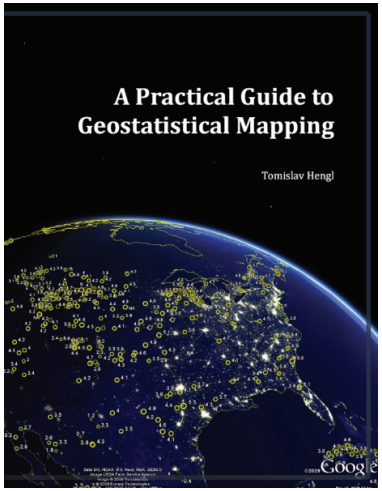
- **Interfaces to other geoinformation software:**
  - Geospatial basics:
    - GDAL: reading/writing of spatial data (package rgdal)
    - Proj4: definition of coordinate reference systems (package rgdal)
  - Geographical Information Systems
    - GRASS GIS (package spgrass6)
    - SAGA GIS (package RSAGA)
    - ArcGIS (via Python with package RPyGeo)
    - PostGIS (packages rgdal or RODBC)
  - GoogleEarth:
    - download of GoogleMaps data (package RgoogleMaps)
    - writing of KML files for GoogleEarth (package maptools, raster)

Literature



Bivand, R, E. Pebesma & V. Gómez-Rubio (2008): Applied Spatial Data Analysis with R. New York: Springer.

- Classes for spatial data in R
- Reading/writing of spatial data
- Mapping
- Point pattern analysis
- Spatial autocorrelation
- Spatial interpolation/Geostatistics
- Disease mapping



Hengl, T. (2009): A Practical Guide to Geostatistical Mapping. Open source book.
http://www.lulu.com/product/file-download/a-practical-guide-to-geostatistical-mapping/14938111

- R as GIS = R + GRASS + SAGA + GoogleEarth
- Geostatistics, spatial autocorrelation, Applications:
    - Geomorphometry, digital elevation models
    - Heavy metal concentrations
    - Land surface temperature

Links

Max Planck Institute for Biogeochemistry

Overview of R packages for spatial data:
http://cran.r-project.org/web/views/Spatial.html

Description of the sp package:
http://cran.r-project.org/web/packages/sp/sp.pdf

Description of the raster package:
http://cran.r-project.org/web/packages/raster/vignettes/Raster.pdf

Some R code examples for spatial data (University Oregon):
http://geography.uoregon.edu/GeogR/index.html

Spatial-analyst.net – a Wiki for spatial data analysis based on R:
http://spatial-analyst.net/

Some more tips:
http://spatialanalysis.co.uk/r/

Mailinglist of R special interest group on using geographical data and mapping
https://stat.ethz.ch/mailman/listinfo/R-SIG-Geo/

Packages for spatial data in R

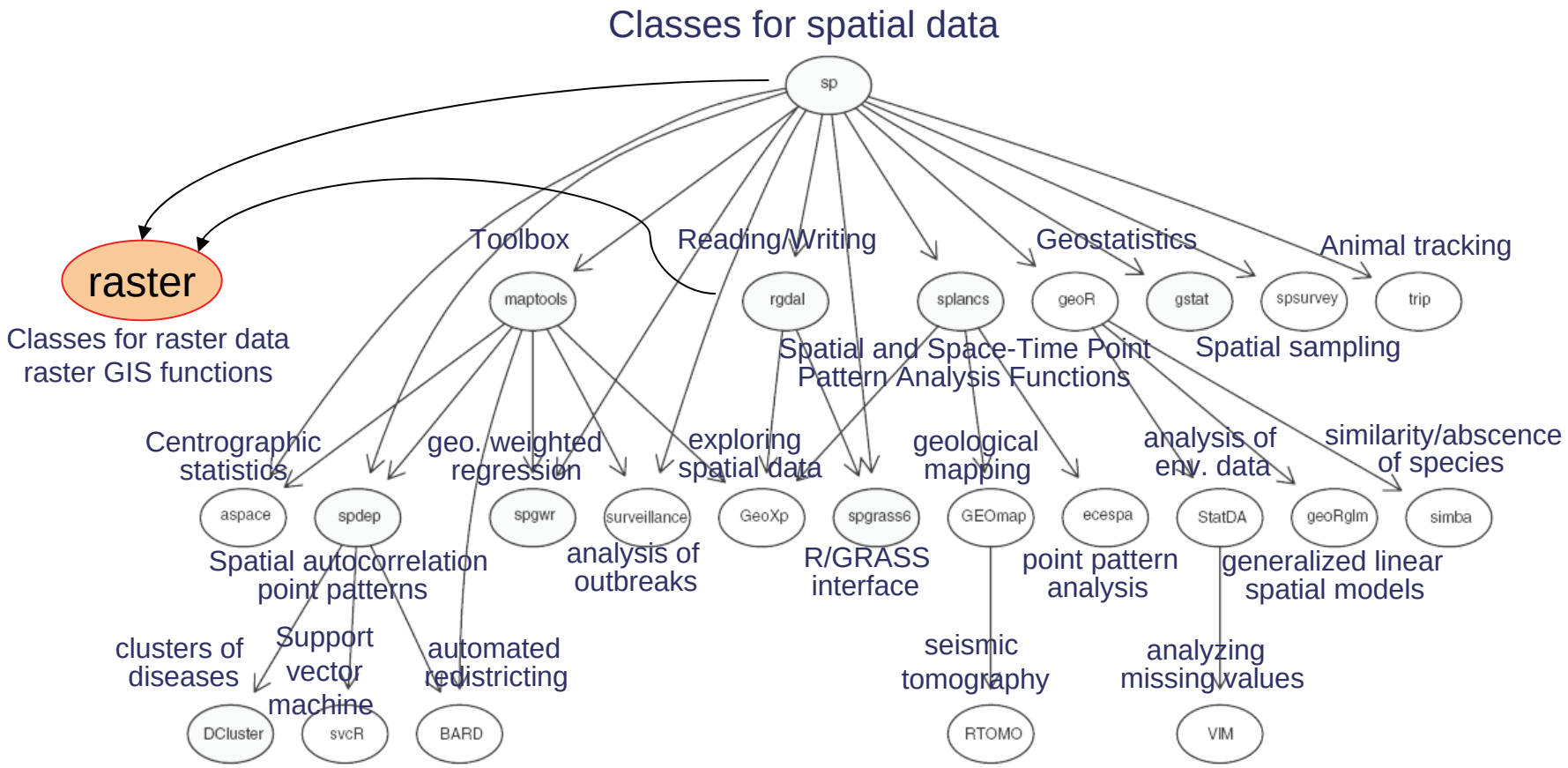


Fig. 1.1. Tree of R contributed packages on CRAN depending on or importing **sp** directly or indirectly; others suggest **sp** or use it without declaration in their package descriptions (status as of 2008-04-06)

Fig.: R packages depending on package sp (Bivand et al. 2008:5)
+ raster package for geographic analysis and modeling with raster data

Exercise 0

Max Planck Institute for Biogeochemistry

Load packages for spatial data in R

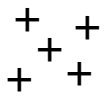
Aim: install and load packages to work with spatial data in R

- Open and run the script 00_spatialR_main.R

Types of spatial data

Vector data

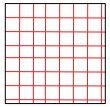
Point



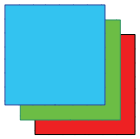
Line

Polygon

Raster data



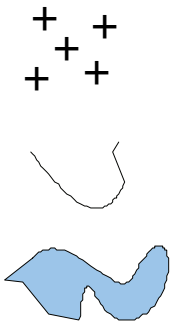
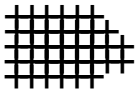
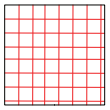
Rasterlayer



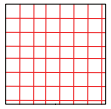
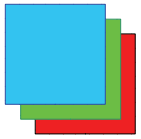
Rasterstack

Classes for spatial data in R

Max Planck Institute for Biogeochemistry

	Data type	read/write	classes from package sp	Basic methods for classes
Vector data		readOGR() writeOGR()	SpatialPoints SpatialPointsDataFrame	Get extent: bbox()
			SpatialLines SpatialLinesDataFrame	Get projection: proj4string()
			SpatialPolygons SpatialPolygonsDataFrame	Get coordinates: coordinates()
			SpatialPixels SpatialPixelsDataFrame	Access data: @data
Raster data		readGDAL() writeGDAL()	SpatialGrid SpatialGridDataFrame	

classes from package raster:

	raster()	RasterLayer
	stack() brick()	RasterStack – multiple files RasterBrick – one file

Get extent: extent()
Get resolution: res()
Get projection: projection()
Get data: getValues()

Classes for vector data: Spatial* classes (package sp)

Max Planck Institute for Biogeochemistry

SpatialPoints, SpatialLines, SpatialPolygons, SpatialPixels

Read ESRI shapefile with country borders:

```
> admin <- readOGR(".", "110m_admin_0_countries")
OGR data source with driver: ESRI Shapefile
Source: ".", layer: "110m_admin_0_countries"
with 177 features and 24 fields
Feature type: wkbPolygon with 2 dimensions
```

```
> bbox(admin)
   min       max
x -180 180.00000
y  -90  83.64513
> proj4string(admin)
[1] " +proj=longlat +ellps=WGS84 +datum=WGS84 +no_def
```

```
> x <- admin@data
> is.data.frame(x)
[1] TRUE
```

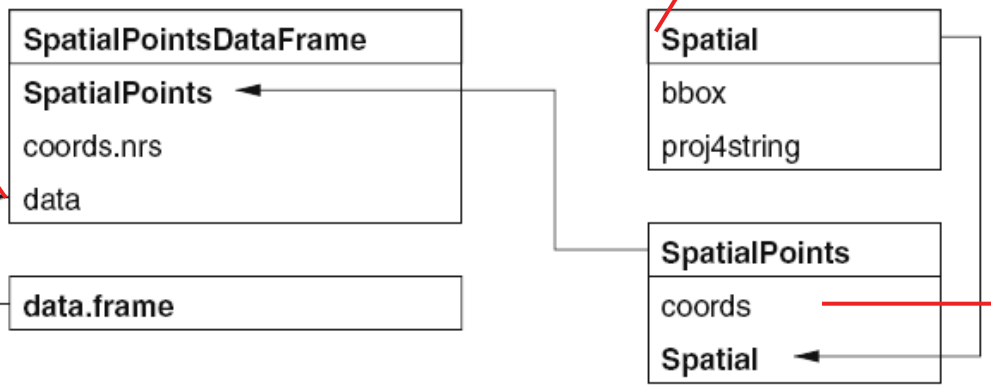


Fig. 2.2. Spatial points classes and their slots; arrows show subclass extensi

```
> coordinates(admin)
             [,1]         [,2]
[1,]   66.086690   33.8563993
[2,]   17.502912  -12.2915534
[3,]   20.032426   41.1413533
[4,]   54.206715   23.8686337
[5,]  -65.149543  -35.2201720
[6,]   45.000290   40.2166076
[7,]   21.284393  -80.5227822
[8,]   69.531580  -49.3064549
```

Fig.: Class hierarchy for Spatial Points (Bivand et al. 2008:35)

Supported file formats: package sp and rgdal*

Max Planck Institute for Biogeochemistry

Raster data: readGDAL(), writeGDAL()

```
> x <- gdalDrivers()
> subset(x, (x$create==TRUE))
       name                                long_name create  copy
2      ADRG        ARC Digitized Raster Graphics   TRUE FALSE
7       BMP MS Windows Device Independent Bitmap   TRUE FALSE
9        BT  VTP .bt (Binary Terrain) 1.3 Format   TRUE FALSE
19     EHdr                    ESRI .hdr Labelled   TRUE  TRUE
21     ELAS                                ELAS    TRUE FALSE
22     ENVI                    ENVI .hdr Labelled   TRUE FALSE
23      ERS                 ERMapper .ers Labelled  TRUE FALSE
34     GSBG   Golden Software Binary Grid (.grd)   TRUE  TRUE
36    GTiff                               GeoTIFF   TRUE  TRUE
38      HFA          Erdas Imagine Images (.img)   TRUE  TRUE
39      IDA               Image Data and Analysis  TRUE FALSE
40    ILWIS                     ILWIS Raster Map   TRUE  TRUE
41     INGR                   Intergraph Raster   TRUE  TRUE
50 Leveller                 Leveller heightfield  TRUE FALSE
51      MEM                    In-memory Raster   TRUE FALSE
52      MFF                    Vexcel MFF Raster  TRUE  TRUE
53      MFF2                  Vexcel MFF2 (HKV) Raster TRUE TRUE
56      NITF National Imagery Transmission Format  TRUE  TRUE
57      PAux                     PCI .aux Labelled  TRUE FALSE
58     PCIDSK                   PCIDSK Database File TRUE  TRUE
62       PNM  Portable Pixmap Format (netpbm)   TRUE FALSE
64       RMF                 Raster Matrix Format  TRUE FALSE
67       RST                    Idrisi Raster A.1  TRUE  TRUE
70       SGI          SGI Image File Format 1.0   TRUE FALSE
72 Terragen                 Terragen heightfield  TRUE FALSE
75       VRT                      Virtual Raster   TRUE  TRUE
```

For working with raster data the classes of the raster package are more feasible …

Vector data:
readOGR(), writeOGR()

```
> ogrDrivers()
                name write
1           AVCBin FALSE
2           AVCE00 FALSE
3              BNA  TRUE
4              CSV  TRUE
5              DGN  TRUE
6  ESRI Shapefile  TRUE
7      Geoconcept  TRUE
8         GeoJSON  TRUE
9              GML  TRUE
10             GMT  TRUE
11             GPX  TRUE
12             KML  TRUE
13   MapInfo File  TRUE
14         Memory  TRUE
15            REC FALSE
16            S57  TRUE
17           SDTS FALSE
18          TIGER  TRUE
19        UK .NTF FALSE
20            VRT FALSE
21         XPlane FALSE
```

* Package rgdal requires GDAL (geospatial data abstraction library) installed (comes e.g. with GRASS, QGIS, FWtools or other FOSS GIS)

Exercise 1

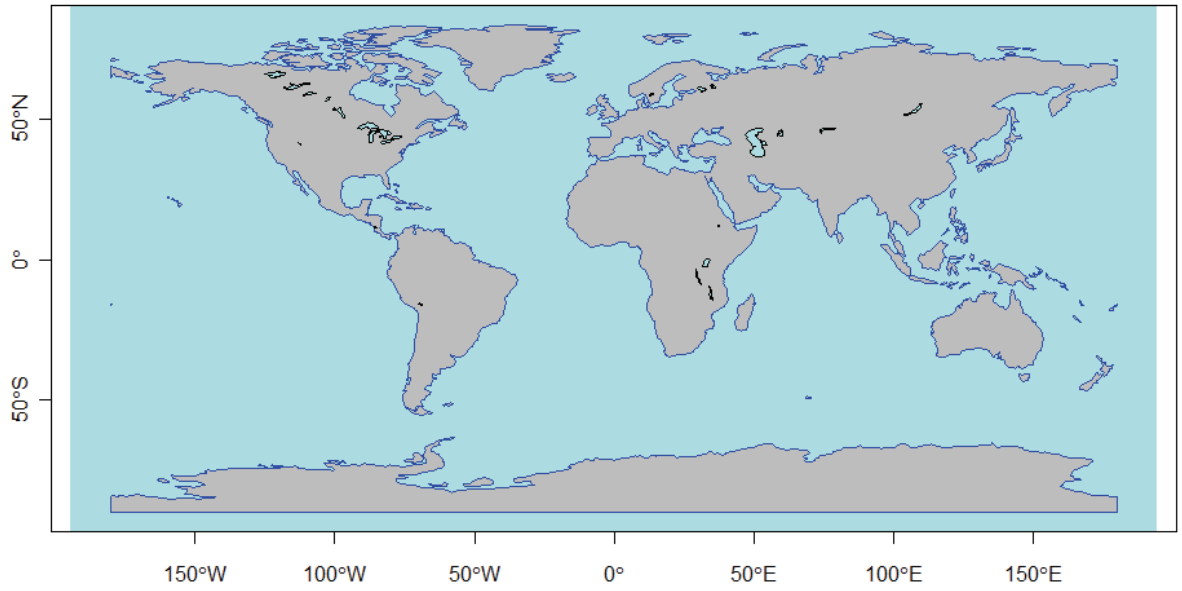
Max Planck Institute for Biogeochemistry

Classes for vector data

Aim: Learn the structure of sp* objects

- Open the script 01_spatialR_ex1_classes-for-vector-data.R
- Run the example

- Exercise 1: make a simple world map



Classes for raster data: RasterLayer, RasterStack, RasterBrick

Max Planck Institute for Biogeochemistry

```
> # data as RasterLayer
> data.rast <- raster("AMSR_E_L3_all_year_2002.tif")
> data.rast
class       : RasterLayer
dimensions  : 132, 1382, 1  (nrow, ncol, nlayers)
resolution  : 25067.53, 25067.53  (x, y)
extent      : -17334194, 17309126, 4010804, 7319717  (xmin, xmax, ymin, ymax)
projection  : +proj=cea +lon_0=0 +lat_ts=30 +x_0=0 +y_0=0 +a=6371228 +b=6371
values      : Z:/R_sp-tut/AMSR_E_L3_all_year_2002.tif
min         : ?
max         : ?

>
> # data as RasterStack
> files <- list("AMSR_E_L3_all_year_2003.tif", "AMSR_E_L3_all_year_2004.tif",
> data.stack <- stack(data.rast, files)
> data.stack
class       : RasterStack
dimensions  : 132, 1382, 4  (nrow, ncol, nlayers)
resolution  : 25067.53, 25067.53  (x, y)
extent      : -17334194, 17309126, 4010804, 7319717  (xmin, xmax, ymin, ymax)
projection  : +proj=cea +lon_0=0 +lat_ts=30 +x_0=0 +y_0=0 +a=6371228 +b=6371
min values  : NA NA NA NA
max values  : NA NA NA NA

>
> # data as RasterBrick
> data.brick <- brick("AMSR_E_L3_all_2002-2008.tif")
> data.brick
class       : RasterBrick
dimensions  : 132, 1382, 7  (nrow, ncol, nlayers)
resolution  : 25067.53, 25067.53  (x, y)
extent      : -17334194, 17309126, 4010804, 7319717  (xmin, xmax, ymin, ymax)
projection  : +proj=cea +lon_0=0 +lat_ts=30 +x_0=0 +y_0=0 +a=6371228 +b=6371
values      : AMSR_E_L3_all_2002-2008.tif
min values  :  6 33 35 41 33 29 17
max values  : 496 414 451 442 451 443 482
```

RasterLayer
- raster()
- only one layer
- reference to file

RasterStack
- stack()
- create from files, RasterLayers or SpatialGridData-Frames
- mutilayer collection of RasterLayers

RasterBrick
- brick()
- multilayer
- data stored in one multiband file
- shortest processing time

(+) data stored on disk, not in memory

Supported file formats: package raster

Max Planck Institute for Biogeochemistry

Read raster data: raster(), stack() or brick()
Write raster data: writeRaster()

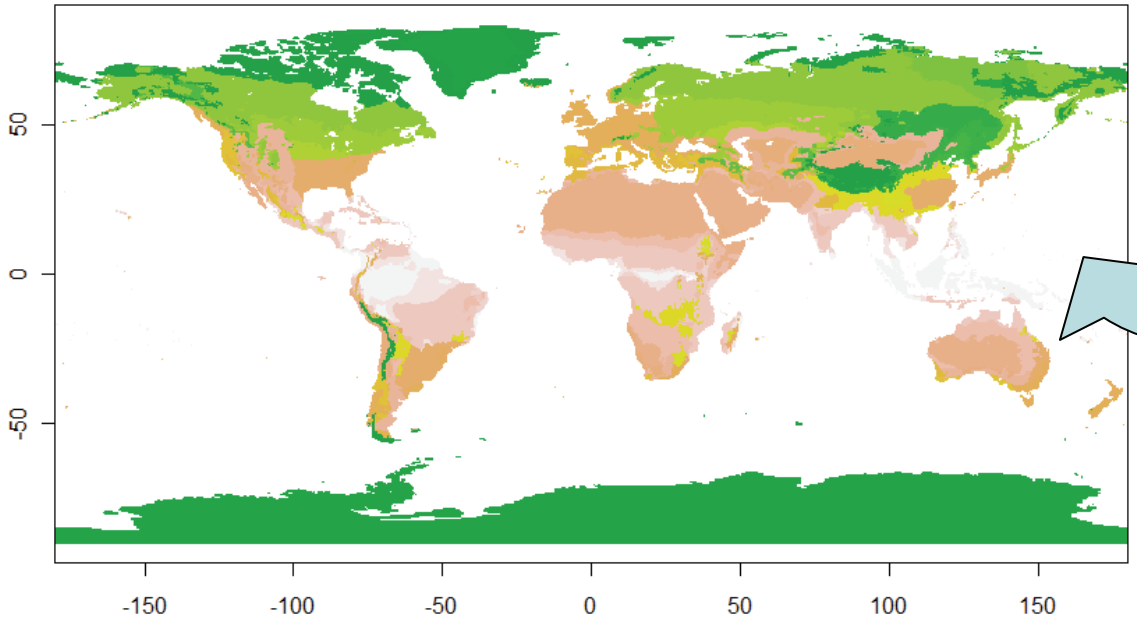
```
> writeFormats()
      name        long_name
 [1,] "raster"    "R-raster"
 [2,] "SAGA"      "SAGA GIS"
 [3,] "IDRISI"    "IDRISI"
 [4,] "BIL"       "Band by Line"
 [5,] "BSQ"       "Band Sequential"
 [6,] "BIP"       "Band by Pixel"
 [7,] "ascii"     "Arc ASCII"
 [8,] "ADRG"      "ARC Digitized Raster Graphics"
 [9,] "BMP"       "MS Windows Device Independent Bitmap"
[10,] "BT"        "VTP .bt (Binary Terrain) 1.3 Format"
[11,] "EHdr"      "ESRI .hdr Labelled"
[12,] "ELAS"      "ELAS"
[13,] "ENVI"      "ENVI .hdr Labelled"
[14,] "ERS"       "ERMapper .ers Labelled"
[15,] "GSBG"      "Golden Software Binary Grid (.grd)"
[16,] "GTiff"     "GeoTIFF"
[17,] "HFA"       "Erdas Imagine Images (.img)"
[18,] "IDA"       "Image Data and Analysis"
[19,] "ILWIS"     "ILWIS Raster Map"
[20,] "INGR"      "Intergraph Raster"
[21,] "Leveller"  "Leveller heightfield"
[22,] "NITF"      "National Imagery Transmission Format"
[23,] "PAux"      "PCI .aux Labelled"
[24,] "PCIDSK"    "PCIDSK Database File"
[25,] "PNM"       "Portable Pixmap Format (netpbm)"
[26,] "RMF"       "Raster Matrix Format"
[27,] "RST"       "Idrisi Raster A.1"
[28,] "SAGA"      "SAGA GIS Binary Grid (.sdat)"
[29,] "SGI"       "SGI Image File Format 1.0"
[30,] "Terragen"  "Terragen heightfield"
```

Exercise 2

Classes for raster data

Aim: learn the structure of raster* objects and create a RasterLayer

- Open the script 02_spatialR_ex2_classes-for-raster-data.R
- Run the example.
- Exercise 2: Create a RasterLayer object



004_gpp-ndvi-comparison_14_compare-trend-method.R

	Lat	Lon	Cls
1			
2	-89.75	-179.75	EF
3	-89.75	-179.25	EF
4	-89.75	-178.75	EF
5	-89.75	-178.25	EF
6	-89.75	-177.75	EF
7	-89.75	-177.25	EF
8	-89.75	-176.75	EF
9	-89.75	-176.25	EF
10	-89.75	5.75	EF
11	-89.75	5.25	EF
12	-89.75	4.75	EF
13	-89.75	4.25	EF
14	-89.75	3.75	EF
15	-89.75	3.25	EF
16	-89.75	-172.75	EF
17	-89.75	-172.25	EF
18	-89.75	-171.75	EF
19	-89.75	-171.25	EF
20	-89.75	-170.75	EF
21	-89.75	-170.25	EF
22	-89.75	-169.75	EF
23	-89.75	-169.25	EF
24	-89.75	-168.75	EF
25	-89.75	-168.25	EF
26	-89.75	-167.75	EF
27	-89.75	-167.25	EF
28	-89.75	-166.75	EF
29	-89.75	-166.25	EF
30	-89.75	-165.75	EF
31	-89.75	-165.25	EF

Mapping

2 Mapping

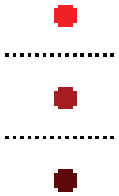
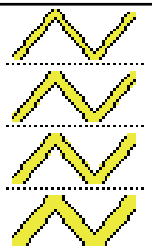
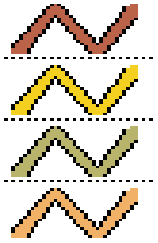
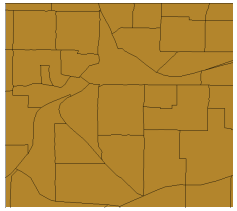
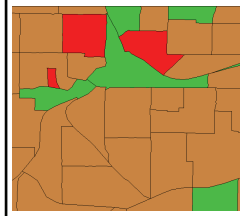
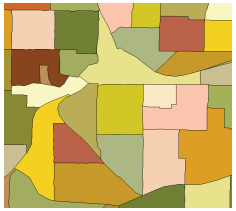
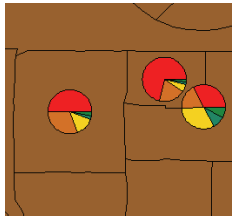
2.1 Plotting functions for spatial data

2.2 Graphical representation of spatial information

2.3 Coordinate systems and reprojection

Graphical representation of vector data (in general, not only R)

Tab.: Legend types for point, line and areal data

Legenden-typ	Single Symbol	Graduated Color	Graduated Symbol	Unique Value	Chart	Dot
Point						--
Line						--
Area			--			

Elements of a map

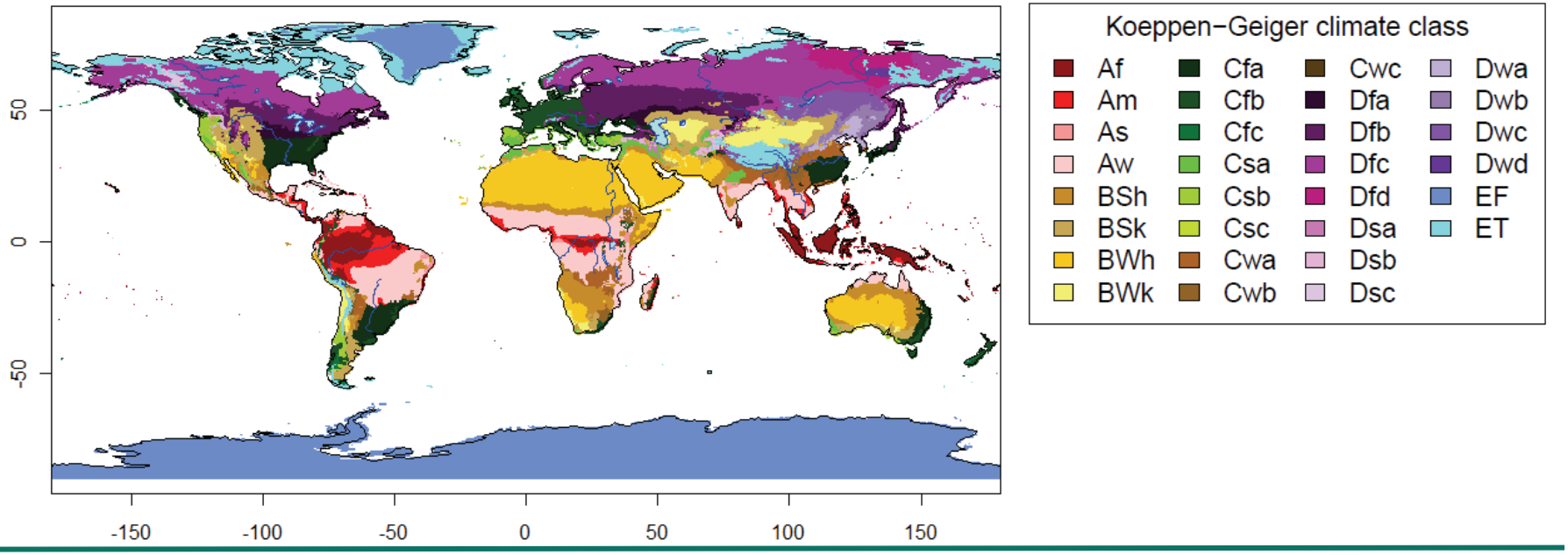
- title / subtitle
- main map, secondary maps
- grid
- Legend explaining colors and symbols
- Info: Projection, geographic reference system
- charts
- references (data sources)
- data type, e.g. information about satellite
- text, explanations
- author, cartographer
- producer, institution, copyright
- scalebar
- north arrow
- map ID (if map belongs to a collection of maps/atlas)
- overview map (for regional maps)

Exercise 3

Create nice maps / plotting functions in R

Aim: create a map including colors, legend and geographical data

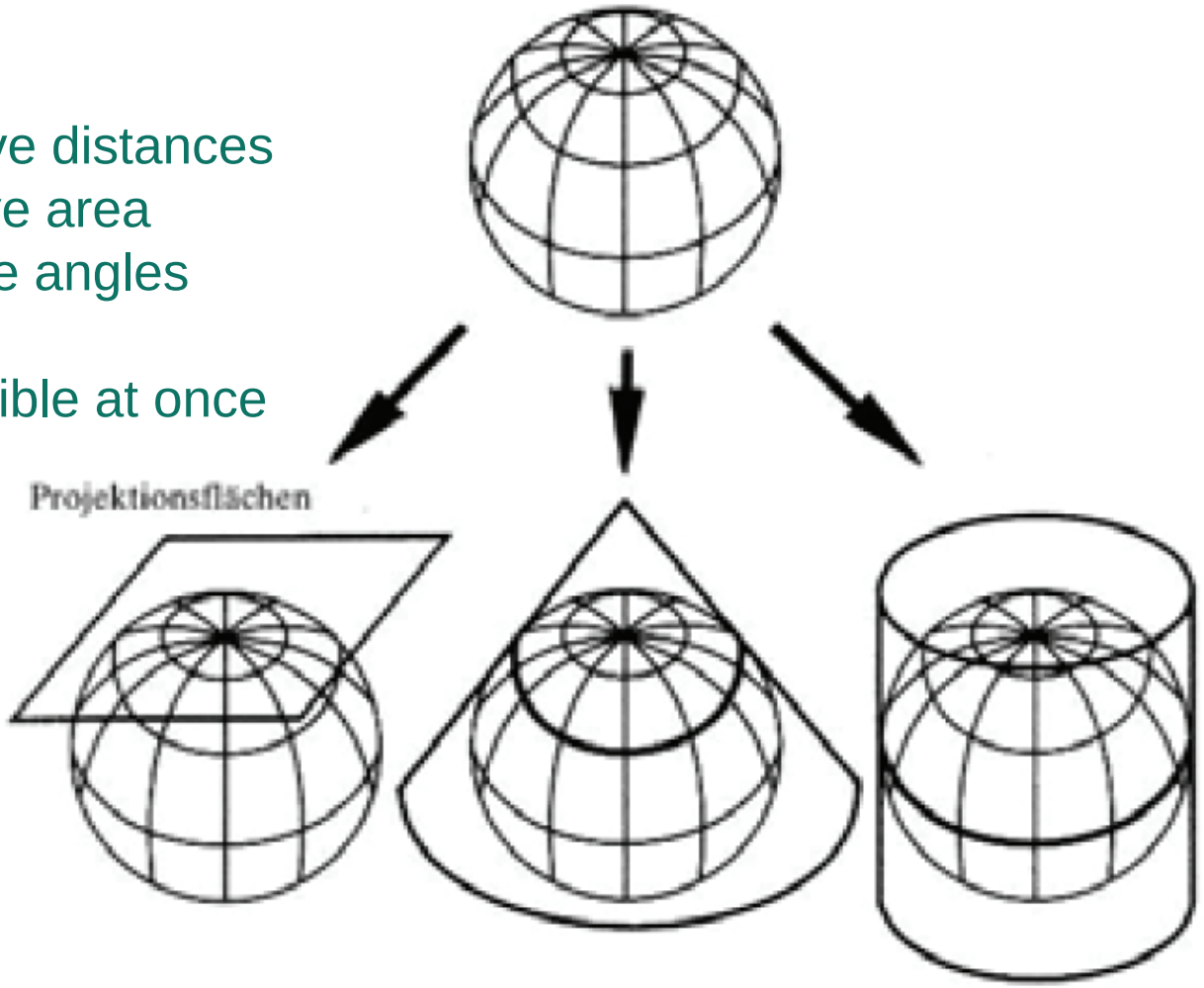
- Open the script 03_spatialR_ex3_mapping1.R
- Run the example.
- Exercise 3: Plot a map of the Koeppen-Geiger climate zones.



Map projections

Properties:
- Equidistant: preserve distances
- Equal area: preserve area
- Conformal: preserve angles

Not everthing is possible at once
(except on a globe)



(aus: Monmonier 1996:25; s. Kohlstock 2004:26)

Proj 4

Max Planck Institute for Biogeochemistry

PROJ.4 - Cartographic Projections Library

→ **http://trac.osgeo.org/proj/**

Standard to define map projections in many open source software.

Examples for common projections:
http://www.remotesensing.org/geotiff/proj_list/

Example for the usual longitude/latitude maps (°E, °N)

```
+proj=longlat +ellps=WGS84 +datum=WGS84 +no_defs +towgs84=0,0,0
```

Projections in R

Max Planck Institute for Biogeochemistry

In R: define the proj4 string as character string and convert it to a CRS (coordinate reference system) object:

```
proj <- CRS(„+proj=longlat +ellps=WGS84
 +datum=WGS84 +no_defs +towgs84=0,0,0“)
```

Projections can be assigned:
… to sp* objects: `proj4string(data) <- CRS(…)`
… to raster* objects: `projection(data) <- CRS(…)`

Reprojection / Spatial transformation:
… for sp* objects: `spTransform(…)`
… for raster* objects:
`projectExtent()`
`projectRaster()`

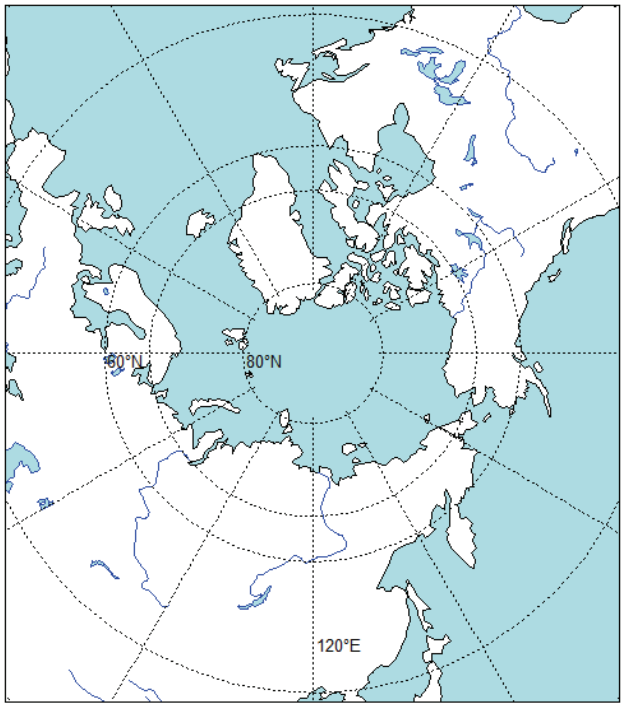
Exercise 4

Max Planck Institute for Biogeochemistry

Transform spatial data

Aim: transform raster and vector data sets (to Robinson projection, Arctic projection), cut a raster with a polygon

- Open the script 04_spatialR_ex4_transformations-mapping2.R
- Run the example.
- Exercise 4: plot a map in an arctic projection



Spatial data analysis

Methods for vector data

Max Planck Institute for Biogeochemistry

Package **sp**: overlay() and over() functions

- Points in polygon
- Grid over polygon

More methods in **rgeos** package

→http://cran.r-project.org/web/packages/rgeos/index.html

e.g. area, distance, buffer, crosses, overlap, intersection, union …

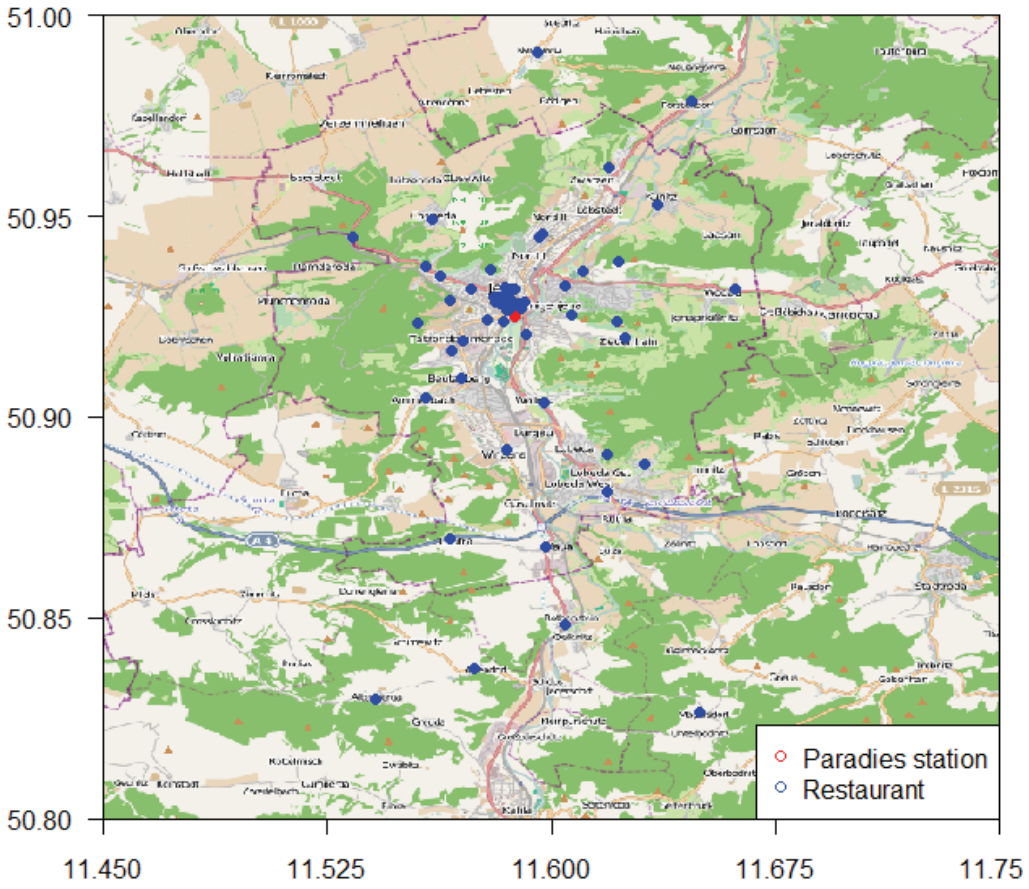
Exercise 7

Max Planck Institute for Biogeochemistry

Spatial analysis with vector data

Aim: make an analysis with point-in-polygon and distance computations

- Open the script 07_spatialR_ex7_spatial-analysis-vector.R
- OpenStreetMap data (downloaded from http://www.geofabrik.de/)
- Run the two examples:
  1. Which landuse type contains most of the benches?
  2. Which is the closest restaurant to the station 'Jena Paradies'?



Methods for raster data

Max Planck Institute for Biogeochemistry

raster package

Description:

http://cran.r-project.org/web/packages/raster/

How to process large rasters (i.e. rasters which cannot be saved in the R memory)?

http://cran.r-project.org/web/packages/raster/vignettes/functions.pdf

GIS methods in the raster package (1)

Max Planck Institute for Biogeochemistry

Resampling:
aggregate() creates raster with lower resolution
disaggregate() creates raster with higher resolution
resample() transfers raster values to different origin and resolution

Spatial transformation:
projectRaster() transforms Raster* objects to another projection
spTransform() transform Spatial* vector objects to other projection

Translocation:
flip() mirrors a raster
rotate() rotates a raster
shift() shifts a raster

Change raster extent:
crop() cuts a raster to a smaller extent
expand() enlarges a raster
merge() merges raster layers to larger extent
mosaic() merges raster layers to larger extent but considers overlap

GIS methods in the raster package (2)

Max Planck Institute for Biogeochemistry

Calculations:

adjacent() identify adjacent cells to a cell on a raster
area() calculates area of raster cells
calc() performs calculations per pixel on multilayer raster objects
hillShade() compute hillshade from slope and aspect
overlay() calculations based on specific layers
terrain() calculate slope and aspect from elevation model
zonal() calculates zonal statistics for a raster

Dealing with NA values:

approxNA() estimate values that are NA in a raster
cover() replaces NA values with values of a second raster
mask() sets values in a raster to NA

Image processing:

edge() detects edges
focal() moving window filters
cut() classify values of a raster to classes
reclass() reclassify values in a raster

GIS methods in the raster package (3)

Max Planck Institute for Biogeochemistry

Raster-vector interfaces:

extract() gets raster values based on other spatial objects
rasterize() converts points, lines or polygons to rasters

Sampling:

sampleRandom() take a random sample from a raster
sampleRegular() take a systematic sample from a raster

Extract values and coordinates:

extract() extract cell values
cellFromCol(), cellFromLine(), cellFromPolygon(), cellFromRow(), cellFromRowCol(), cellFromXY(), cellFromExtent() get cell number from a raster
xFromCell(), yFromCell(), xyFromCell() get coordinates from cell number
minValue(), maxValue() extreme values in a raster

Plotting/Mapping:

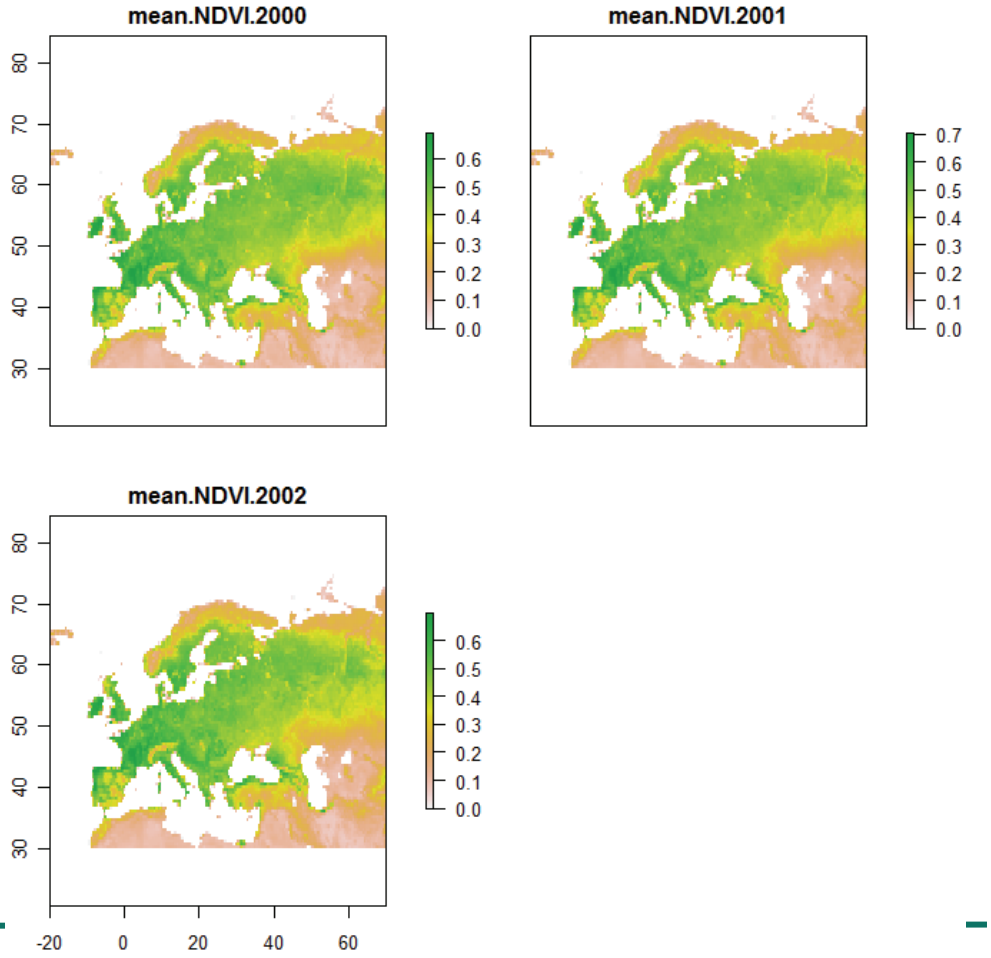
plot() plot a map of a Raster* object
plotRGB() plot red-green-blue composite of a raster
contour() plot contour lines
pairs() plot matrix of scatterplots for layers in a RasterBrick or RasterStack
scalebar() add a scalebar to a raster map

Exercise 8

Calculations with raster data

Aim: learn statistical and arithmetical functions for raster data, analyse time series of raster data

- Open the script 08_spatialR_ex8_spatial-analysis-raster.R
- Run the example.
- Exercise 8: Compute zonal statistics of NDVI by Koeppen-Geiger climate zones



Spatial R and GoogleEarth

4 Exchange between R and other spatial software

4.1 Import of GoogleMaps and OpenStreetMap data in R

4.2 Export data to GoogleEarth

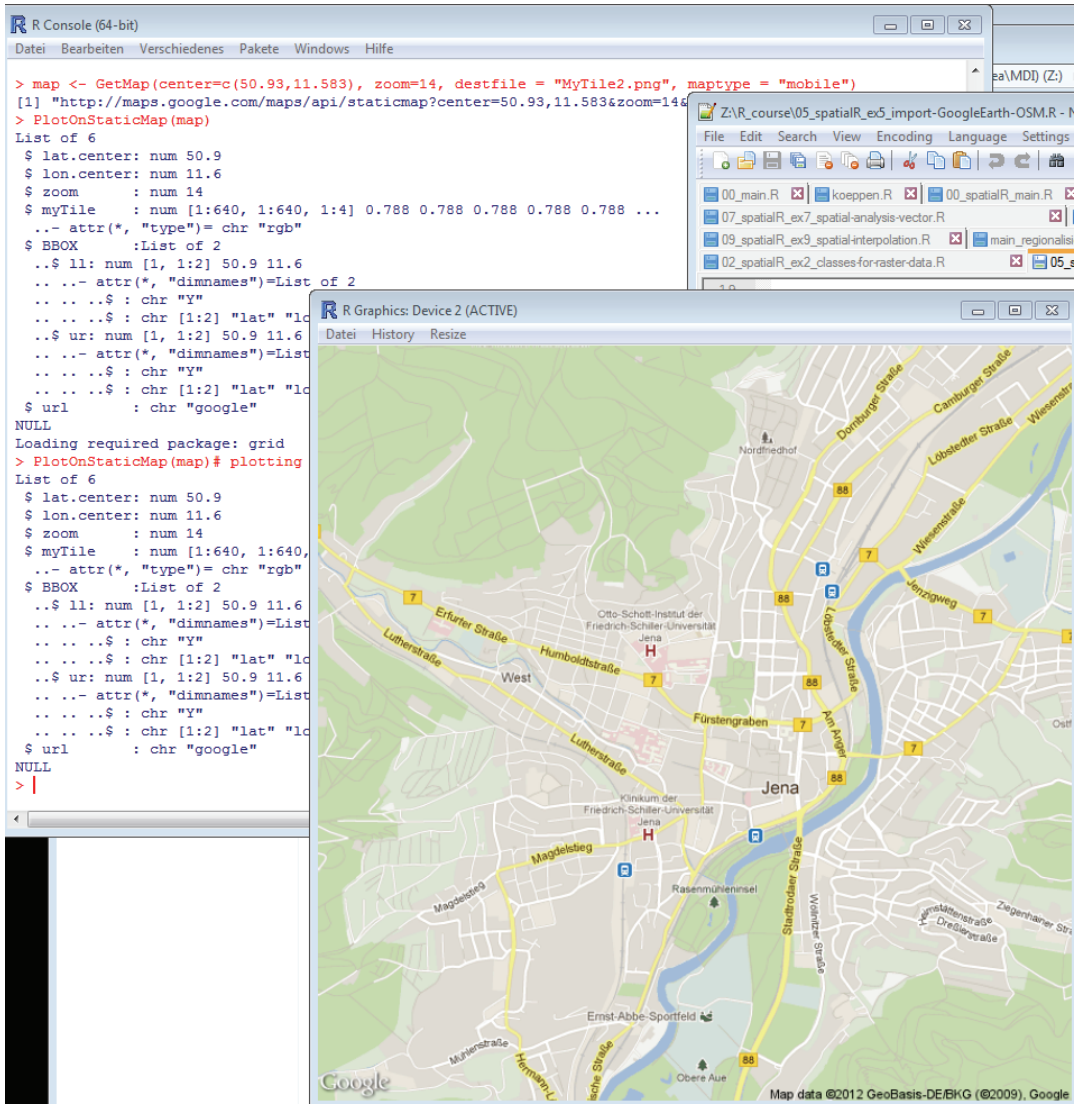
4.3 Linking R with GIS software

RgoogleMaps

Max Planck Institute for Biogeochemistry

Download data from GoogleMaps and OpenStreetMap with RgoogleMaps package

http://www.openstreetmap.org



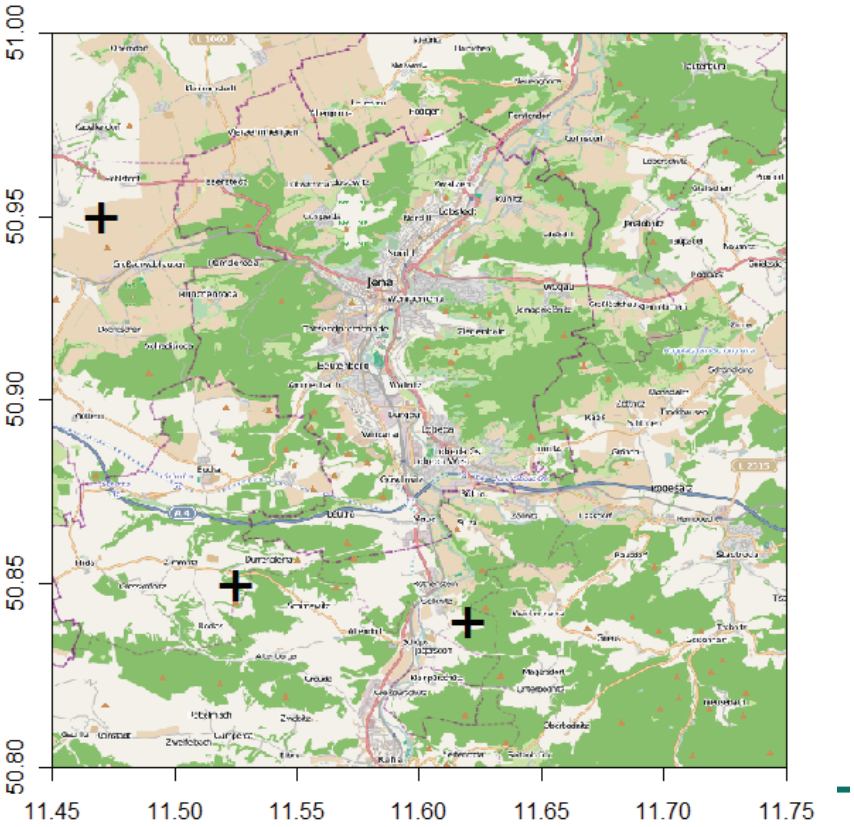
Exercise 5

Max Planck Institute for Biogeochemistry

Import data from GoogleMaps and OpenStreetMap to R

Aim: Learn how to use spatial data from web services for your own maps

- Open the script 05_spatialR_ex5_import-GoogleEarth-OSM.R
- Run the demonstration.

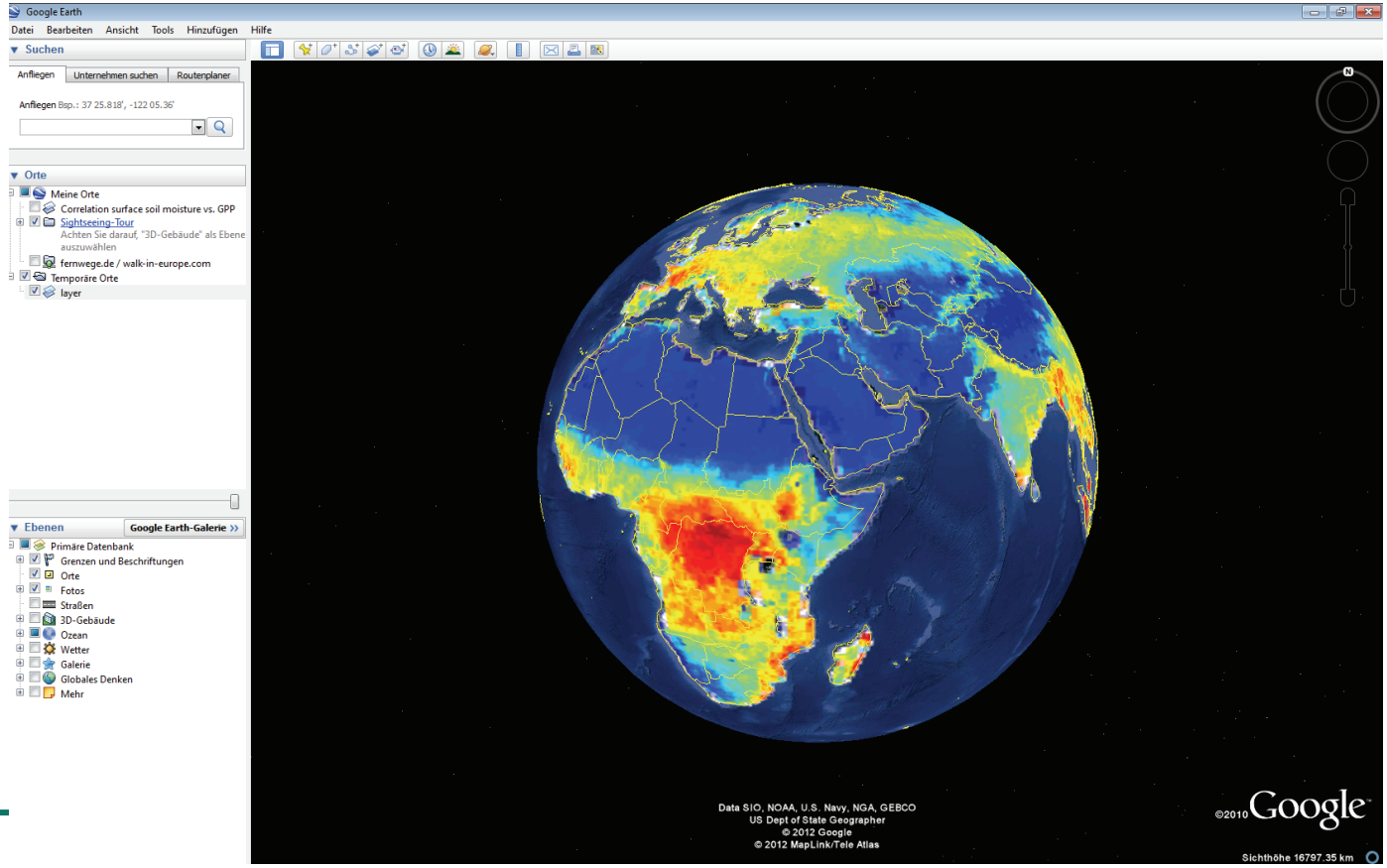


Exercise 6

Export your own data to GoogleEarth

Aim: Create a kml-file from a raster for an overlay in GoogleEarth

- Open the script 06_spatialR_ex6_export-GoogleEarth.R
- Run the demonstration.



R and GIS

Max Planck Institute for Biogeochemistry

How to access functionality of other GIS software from R?

GRASS GIS: package spgrass6

ArcGIS: via package RPyGeo

SAGA GIS: package RSAGA (wraps SAGA shell commands in R functions)

QGIS: you can use R within QGIS