A roller coaster introduction to spatial analysis using R

Photo by Mark Asthoff on Unsplash

Stephanie Kramer-Schadt (IZW)

The teachers

Ana Patricia Calderón

Research interests:

- Carnivore/felid conservation
- Connectivity
- Biological corridors

DAAD fellow and Doctoral candidate:

- Connectivity of jaguar populations in Central America
- Universität Potsdam - UFZ – Leibniz IZW; associated with BioMove
- *Aim of project:* improve the connectivity of jaguar populations in Central America to secure their long term viability

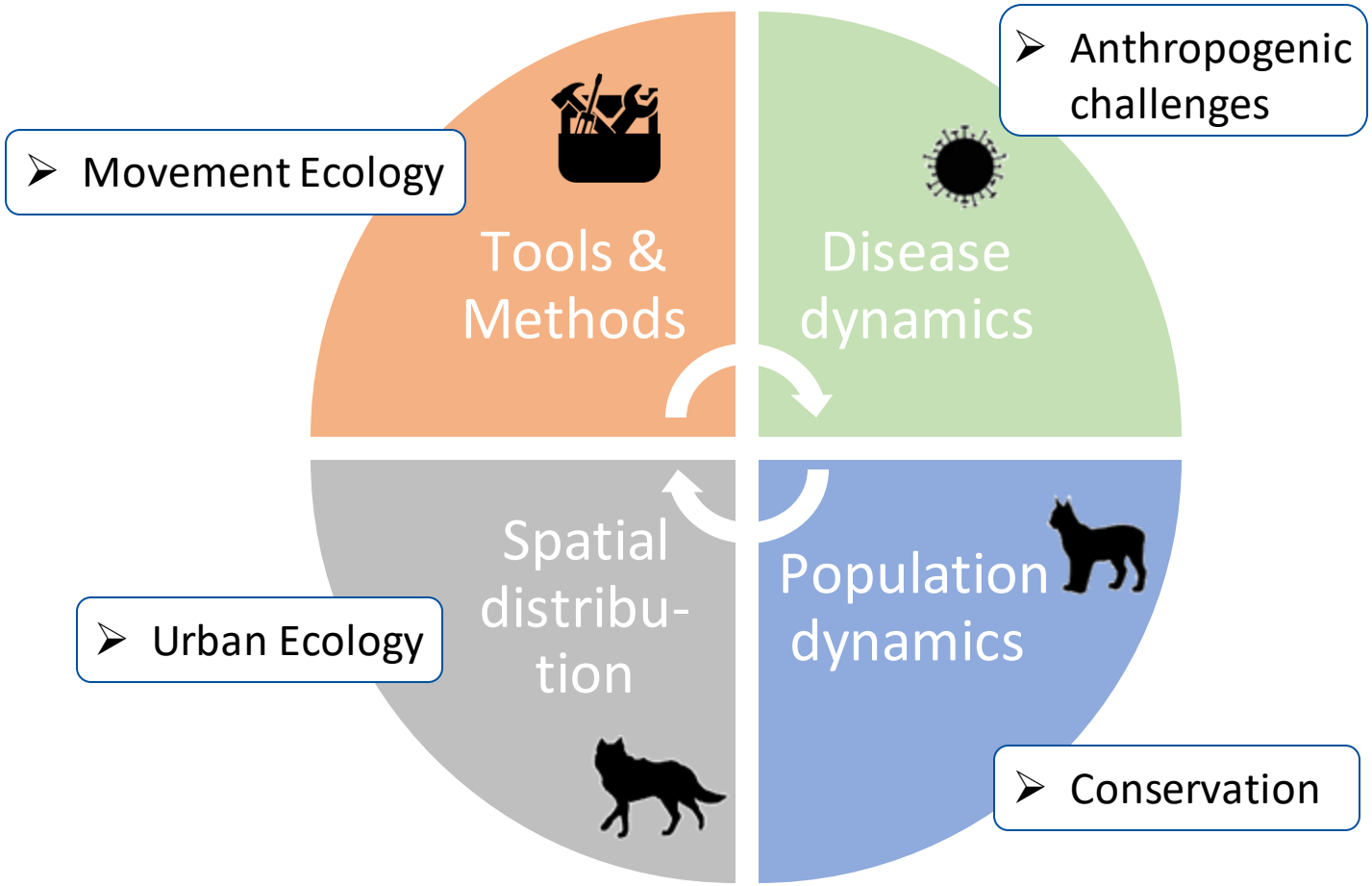
The teachers

Cédric Scherer

- PhD in Ecology 2019
  Movement and Disease Ecology
  Simulation Models + Analysis
- PostDoc at IZW since 2019
  Computational Ecology
  Data Analysis + Visualization,
  GIS Data, Lab Support
- Independent DataViz Specialist since 2020
  Consulting, Design + Coaching

The teachers

Stephanie Kramer



Aim of the Course

To get you going doing spatial operations and analyses in R

- To understand the principle of a GIS (structure, layers,…)
- To see GIS as a tool for analysis
- To be able to visualize and analyze own field data

- To exercise a lot so that loading and manipulating spatial data gets a daily routine

TAKE HOME MESSAGE: spatial R is the same as handling basic data types (data.frame, matrix, list) – just that you need to understand the added spatial component handling (*geometry*)

Outline – spatial analysis in R

DAY 1

LECTURE (Steph)

- Introduction – What is a GIS?
- Coordinate reference systems, Projections & Analyses
- Data types: vector (shapefiles) and raster data
- Migrating from GUI-GIS to spatial R (basic packages)

QUESTIONS to pre-course EXERCISES 1 (Patricia)

- First Steps with Loading spatial data / Handling data types in R / Layout

TIPS‘n TRICKS (Steph / Cedric)

- RMarkdown

EXERCISES 2 (& 3) (Cedric / Steph + Patricia)

- DIY – and use package d6Berlin

Breakout rooms in Zoom
Work in groups

Disclaimer

- A GIS does not free you from thinking!
- Definition of the problem --> User!
- The results are only as good as the question, data and analyses! -> this course does not cover statistics!

Please ask if you do not understand s.th.!

At any time!

So confusing, she couldn‘t explain anything…

Yeah, she spoke way too fast!

That was terrible!

Buh buh

https://en.wikipedia.org/wiki/File:Statler_and_Waldorf.jpg#/media/File:Statler_and_Waldorf.jpg

What is a GIS?

= Geographical Information System

→ A system, that is able to display, save, edit, analyze and govern spatial data.

→ E.g. used for remote sensing, geography, urban and landscape planning…

→ Frequently used systems:
ESRI ArcGIS©,
GRASS, R, QGIS, SAGA (open source!),
ERDAS Imagine, …

http://en.wikipedia.org/wiki/Geographic_information_system

Historic Cartography

Map of Europe of Mercator

Sea travel 15th/16th century

e.g. Gerhard Mercator

Developped a projection in 1569 that is important for sea travel (and aviation) till this day due to its high degree of conformity – the Mercator projection

http://de.wikipedia.org/wiki/Gerhard_Mercator

First ‚Application' (as Analysis Tool)

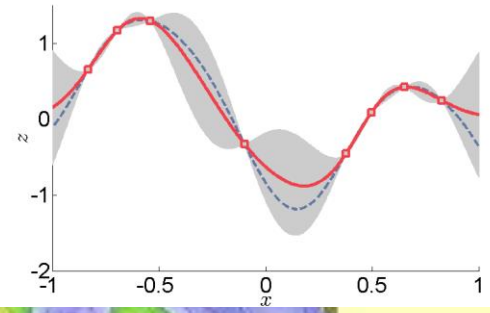
- 1854 John Snow
- Onset of cholera in London and location of water pumps
  - → spatial cluster analysis
  - → foundation of epidemiology as a science

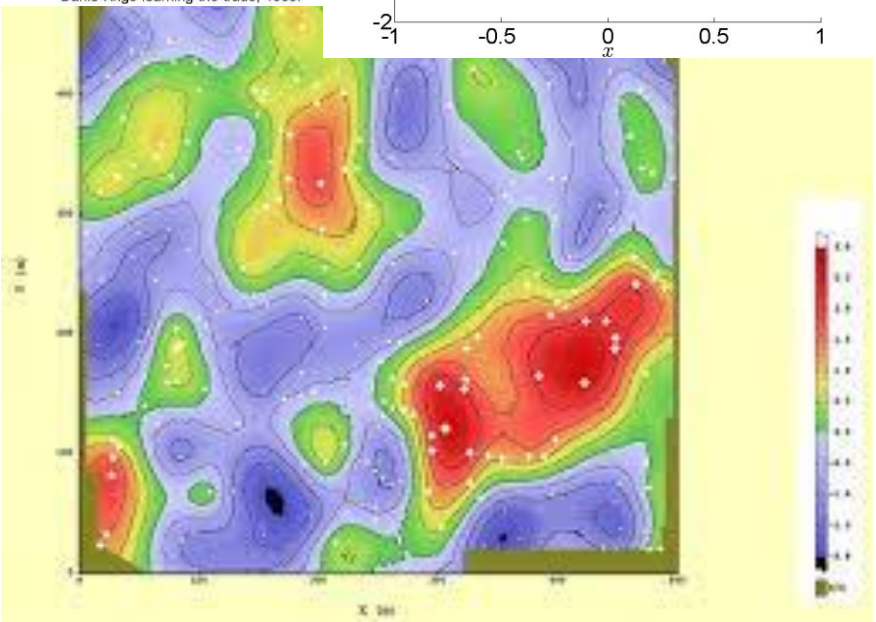
First uses of map-based spatial analysis

http://en.wikipedia.org/wiki/John_Snow_(physician)

‚Kriging' – spatial interpolation

Danie Krige learning the trade, 1939.



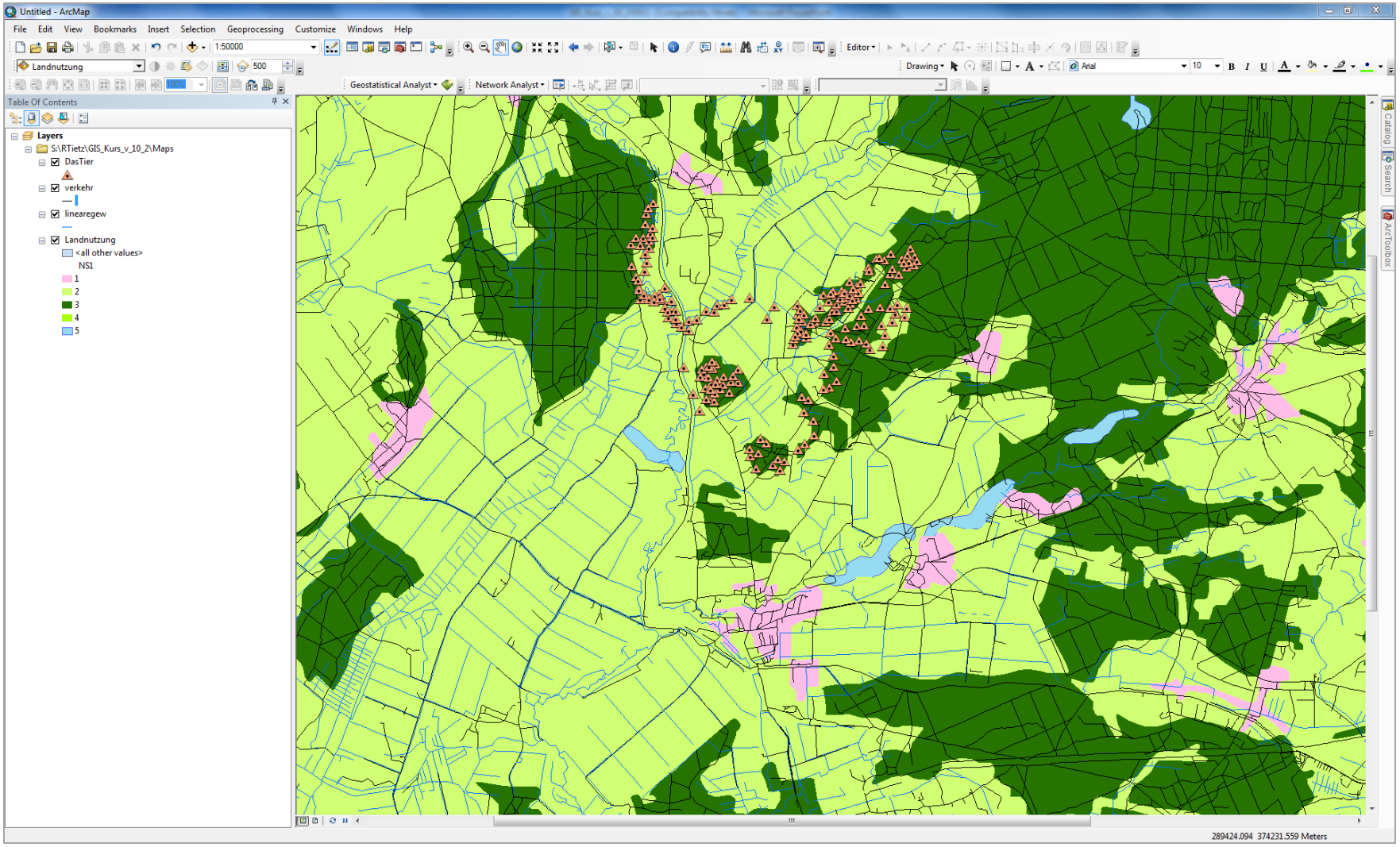


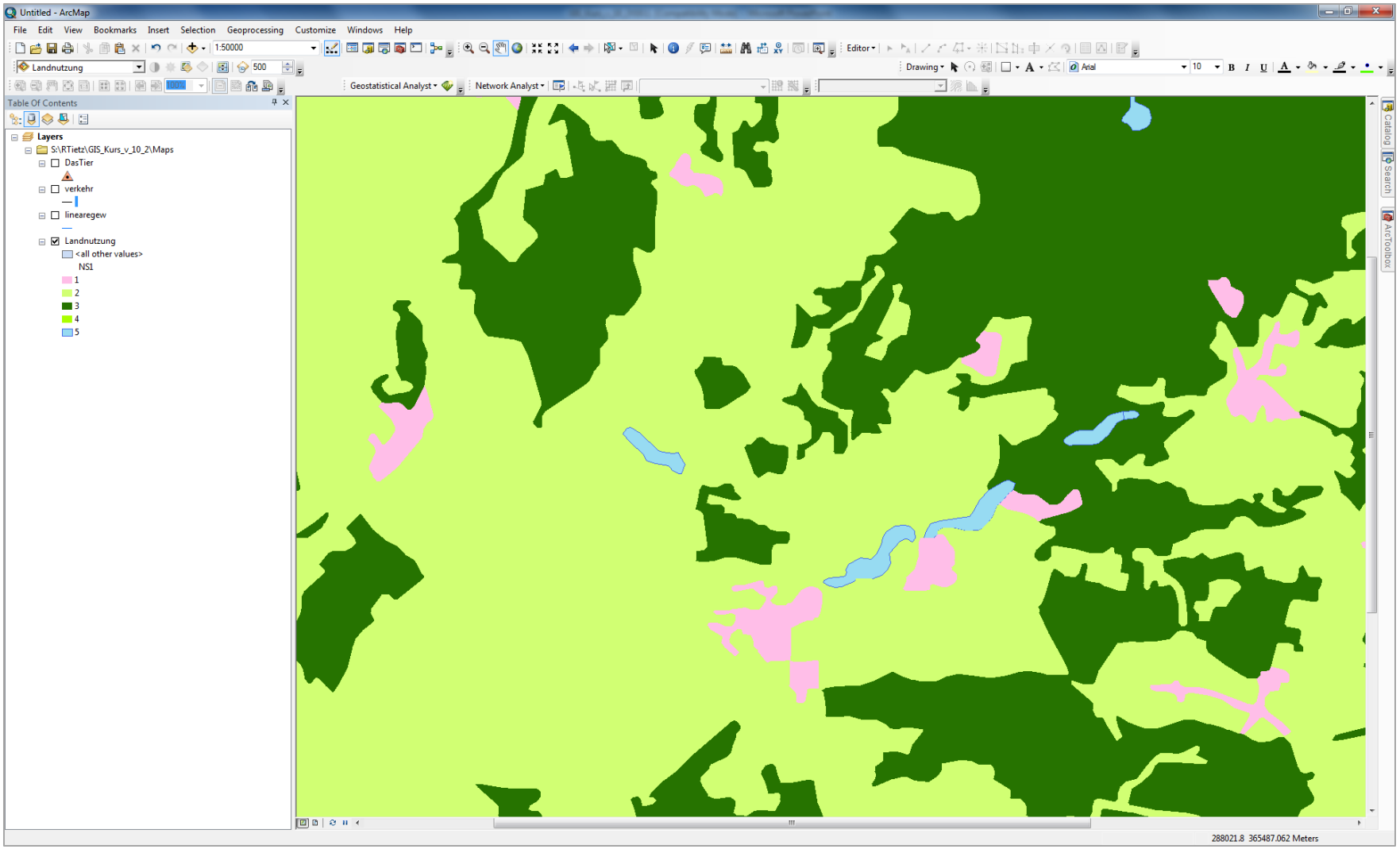
- Daniel G. Krige (*1919-)
- South African mining engineer

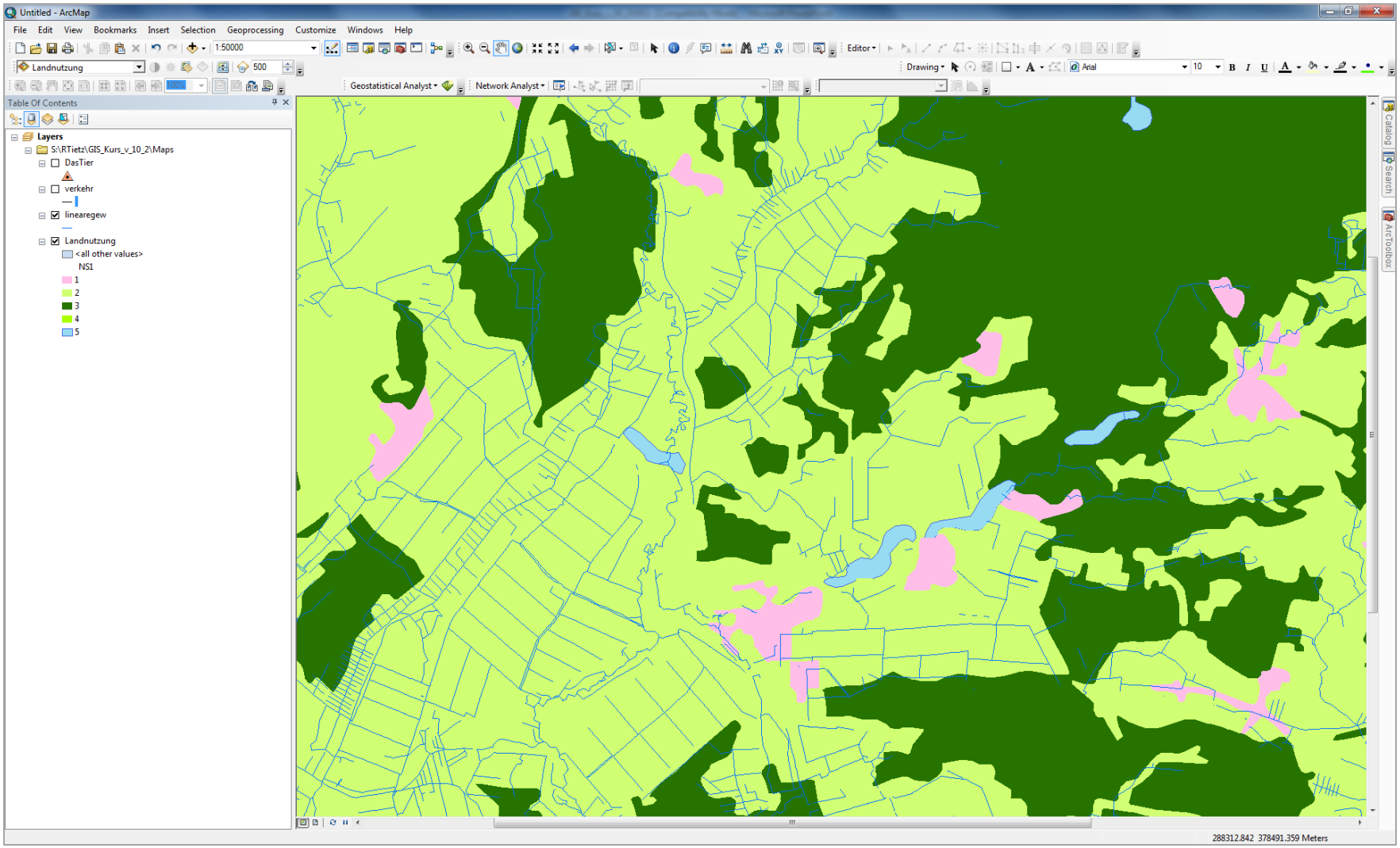
- An interpolation technique
- surrounding measured values are weighted to derive a predicted value for an unmeasured location.
- Weights are based on the distance between the measured points, the prediction locations, and the overall spatial arrangement among the measured points.

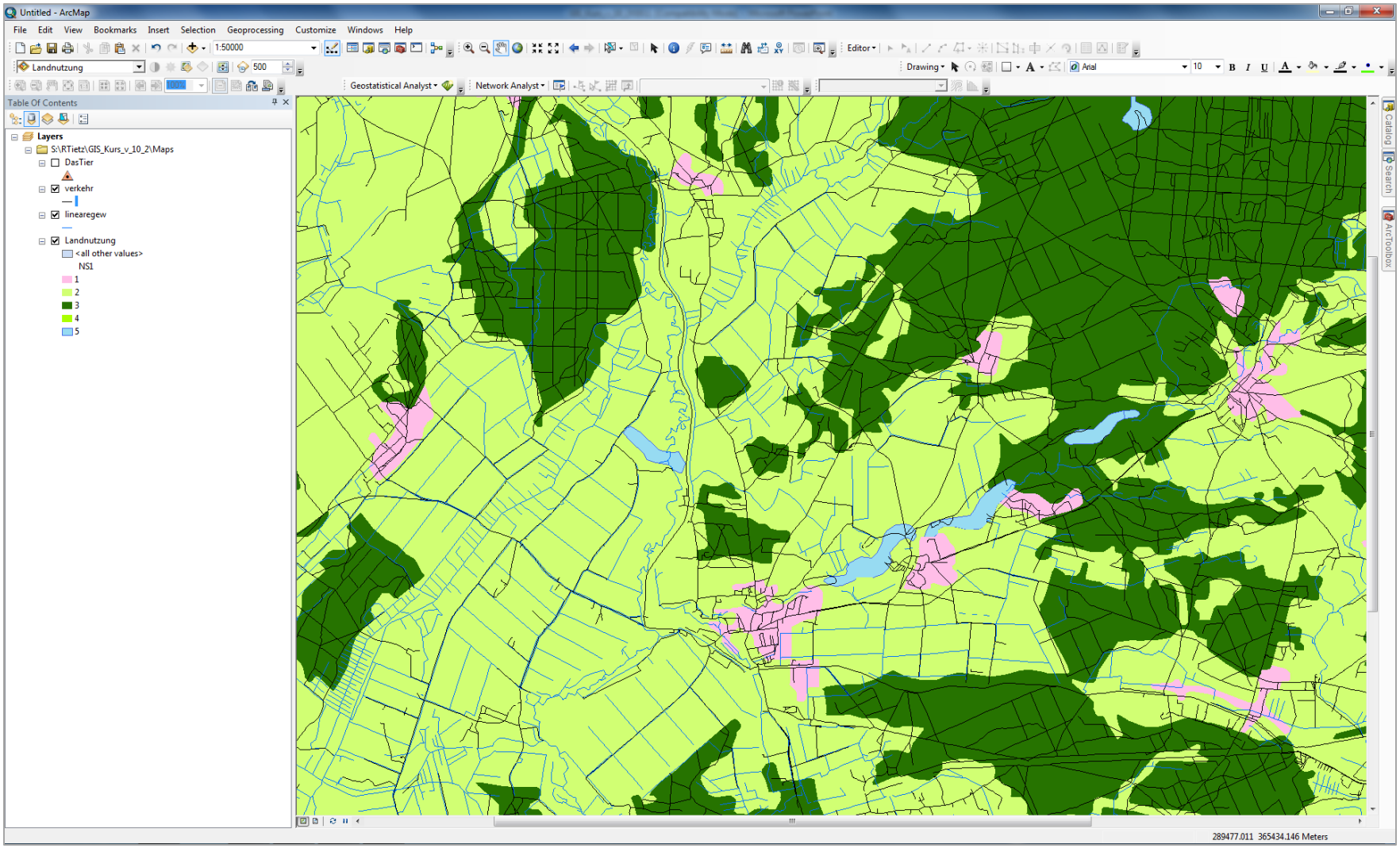
http://www.gislounge.com/danie-krige-kriging; https://en.wikipedia.org/wiki/Spatial_analysis

GIS… Just a Digital Map?

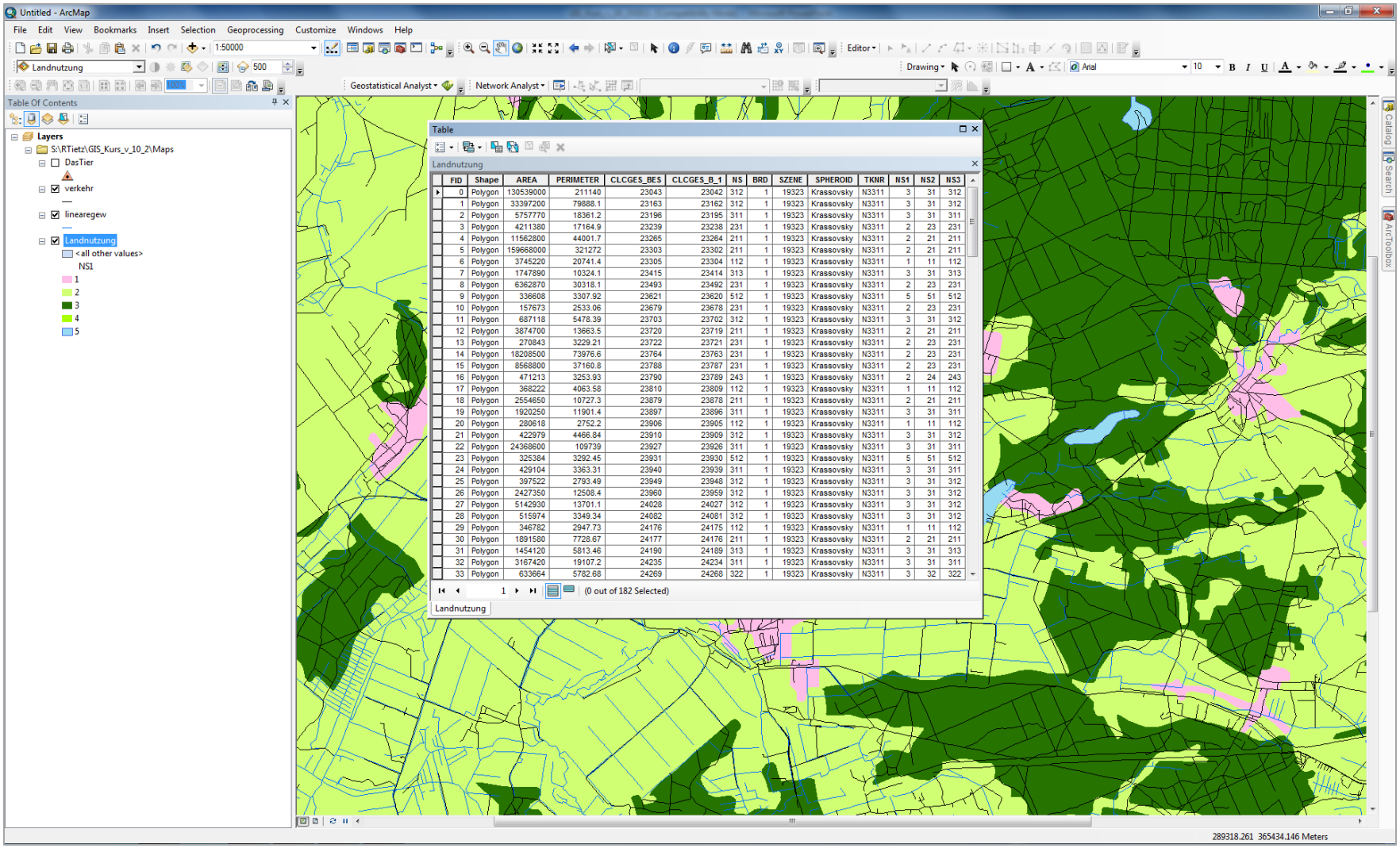



Untitled - ArcMap
File Edit View Bookmarks Insert Selection Geoprocessing Customize Windows Help
Table Of Contents
Layers
S:\RTietz\GIS_Kurs_v_10_2\Maps
DasTier
verkehr
linearegew
Landnutzung
<all other values>
NS1
1
2
3
4
5
Catalog
Search
ArcToolbox
288021.8 365487.062 Meters


Untitled - ArcMap
File Edit View Bookmarks Insert Selection Geoprocessing Customize Windows Help
Table Of Contents
Layers
S:\RTietz\GIS_Kurs_v_10_2\Maps
DasTier
verkehr
linearegew
Landnutzung
<all other values>
NS1
1
2
3
4
5
Geostatistical Analyst
Network Analyst
Drawing
Editor
Catalog
Search
ArcToolbox
288312.842 378491.359 Meters

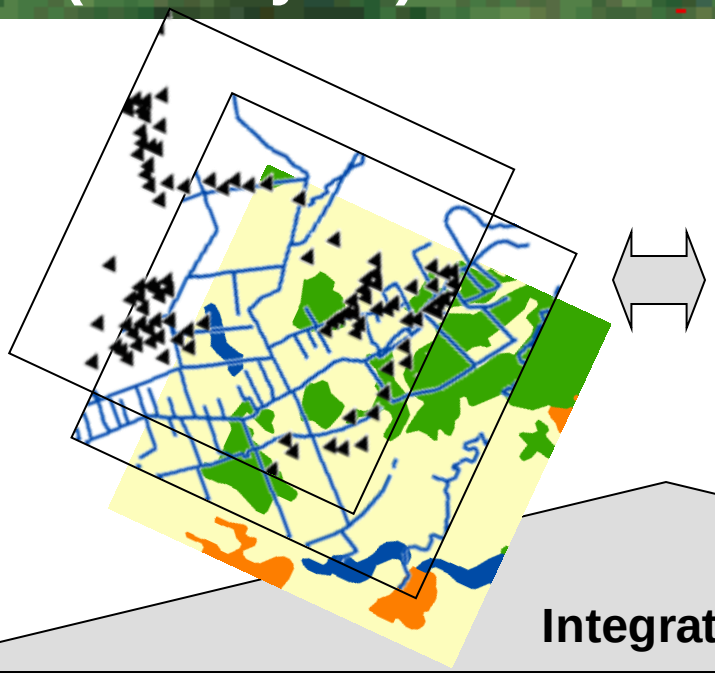

Untitled - ArcMap
File Edit View Bookmarks Insert Selection Geoprocessing Customize Windows Help
Table Of Contents
Layers
S:\RTietz\GIS_Kurs_v_10_2\Maps
DasTier
verkehr
linearegew
Landnutzung
<all other values>
NS1
1
2
3
4
5
Geostatistical Analyst
Network Analyst
Drawing
Editor
Catalog
Search
ArcToolbox
289477.011 365434.146 Meters

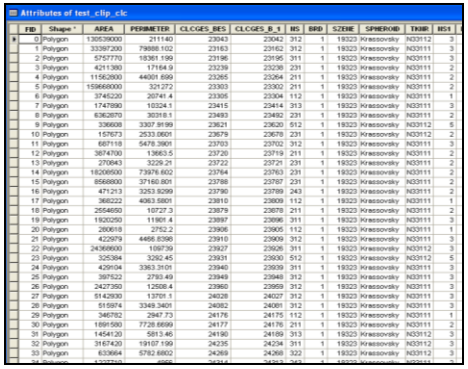
...or a Database?



Both! GIS: spatial layers + database

(+ analysis)





Attributes of test_clip_clc

FID	Shape *	AREA	PERIMETER	CLCGES_BES	CLCGES_B_1	NS	BRD	SZENE	SPHEROID	TKNR	RS1
0	Polygon	130539000	211140	23043	23042	312	1	19323	Krassovsky	N33112	3
1	Polygon	33397200	79888.102	23163	23162	312	1	19323	Krassovsky	N33111	3
2	Polygon	5757770	18361.199	23196	23195	311	1	19323	Krassovsky	N33111	3
3	Polygon	4211390	17164.9	23239	23238	231	1	19323	Krassovsky	N33111	2
4	Polygon	11562800	44001.699	23265	23264	211	1	19323	Krassovsky	N33111	2
5	Polygon	159669000	321272	23303	23302	211	1	19323	Krassovsky	N33111	2
6	Polygon	3745220	26741.4	23305	23304	112	1	19323	Krassovsky	N33111	1
7	Polygon	1747890	10324.1	23415	23414	313	1	19323	Krassovsky	N33111	3
8	Polygon	6362870	30318.1	23493	23492	231	1	19323	Krassovsky	N33111	2
9	Polygon	336609	3307.9199	23621	23620	512	1	19323	Krassovsky	N33112	5
10	Polygon	157673	2533.8601	23679	23679	231	1	19323	Krassovsky	N33112	2
11	Polygon	687118	5478.3901	23703	23702	312	1	19323	Krassovsky	N33111	3
12	Polygon	3674700	13663.5	23720	23719	211	1	19323	Krassovsky	N33111	2
13	Polygon	270943	3229.21	23722	23721	231	1	19323	Krassovsky	N33111	2
14	Polygon	18206500	73976.602	23764	23763	231	1	19323	Krassovsky	N33111	2
15	Polygon	8568800	37160.801	23788	23787	231	1	19323	Krassovsky	N33111	2
16	Polygon	471213	3253.8299	23790	23789	243	1	19323	Krassovsky	N33111	2
17	Polygon	368222	4063.5801	23810	23809	112	1	19323	Krassovsky	N33111	1
18	Polygon	2554650	10727.3	23879	23878	211	1	19323	Krassovsky	N33111	2
19	Polygon	1920250	11901.4	23897	23896	311	1	19323	Krassovsky	N33111	3
20	Polygon	280618	2752.2	23906	23905	112	1	19323	Krassovsky	N33111	1
21	Polygon	422979	4466.8398	23910	23909	312	1	19323	Krassovsky	N33111	3
22	Polygon	2436900	10873.9	23927	23926	311	1	19323	Krassovsky	N33112	3
23	Polygon	325384	3292.45	23931	23930	512	1	19323	Krassovsky	N33112	5
24	Polygon	429104	3363.3101	23940	23939	311	1	19323	Krassovsky	N33111	3
25	Polygon	387522	2793.49	23949	23948	312	1	19323	Krassovsky	N33111	3
26	Polygon	2427350	12508.4	23960	23959	312	1	19323	Krassovsky	N33111	3
27	Polygon	5142930	13701.1	24028	24027	312	1	19323	Krassovsky	N33111	3
28	Polygon	515974	3349.3401	24082	24081	312	1	19323	Krassovsky	N33111	3
29	Polygon	346782	2947.73	24176	24175	112	1	19323	Krassovsky	N33111	1
30	Polygon	1891580	7728.6699	24177	24176	211	1	19323	Krassovsky	N33111	2
31	Polygon	1454120	5813.46	24190	24189	313	1	19323	Krassovsky	N33112	3
32	Polygon	3167420	19107.199	24235	24234	311	1	19323	Krassovsky	N33112	3
33	Polygon	633664	5782.6802	24269	24268	322	1	19323	Krassovsky	N33112	3

Integration

Geographic Information

- Points
- Lines
- Polygons
- (Topography 3D)

Factual information eg:

- building types/ tree types
- road types/ river categories
- cities, inhabitant density
- elevation, precipitation

…a Georeferenced Database

A GIS consists of four functional components:

acquisition, **maintenance**, **analysis**, **presentation**

Acquisition is the digitization of geoobjects, the input of attribute data and the import of external data, e.g. existing maps, databases or graphics.

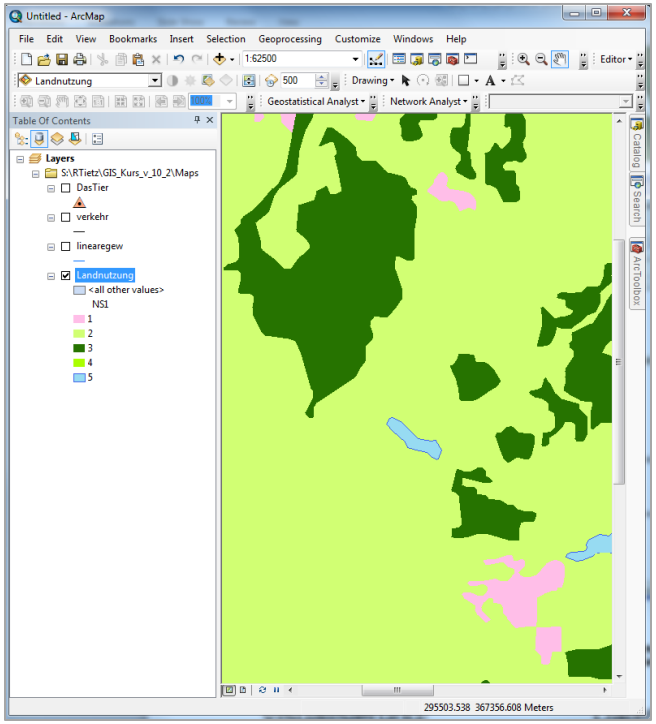
Maintenance is the development and organization of the databases for attributes and geometry of the geoobjects.

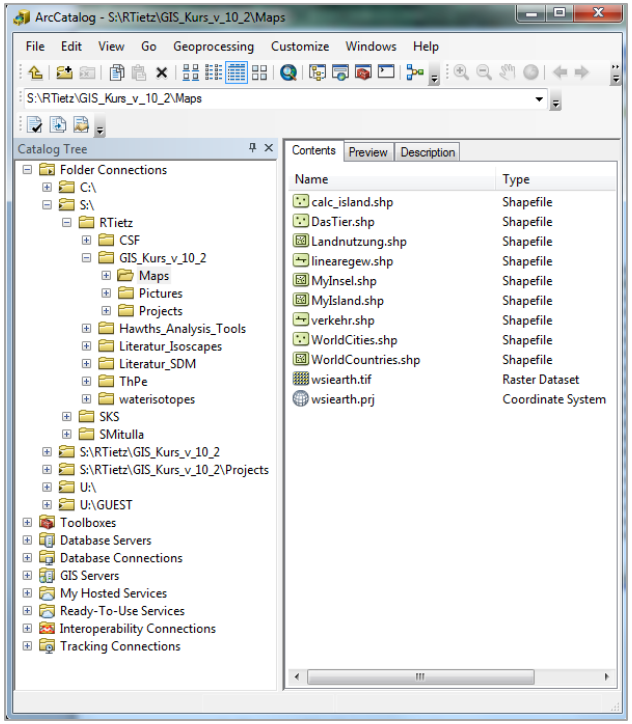
…a Georeferenced Database

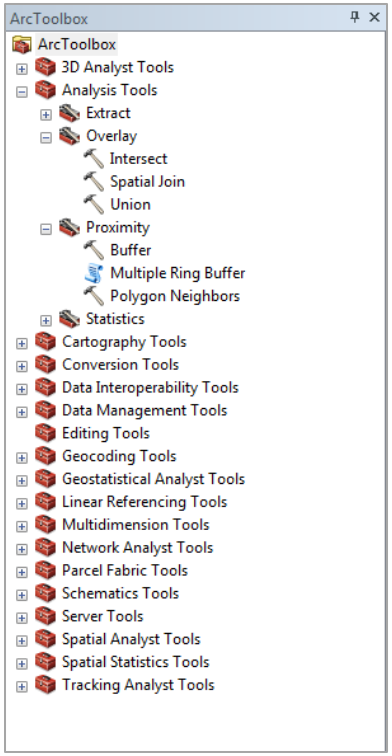
The **Analysis** consist of the deduction of new information through logic and geometric links and scanning.

Finally, the **Presentation** displays information in maps and is complemented by texts, diagrams, tabels, legends, scales, north arrow etc.

Program Structure of ArcGIS© ESRI







ArcMap

Visualization/
Presentation

ArcCatalog

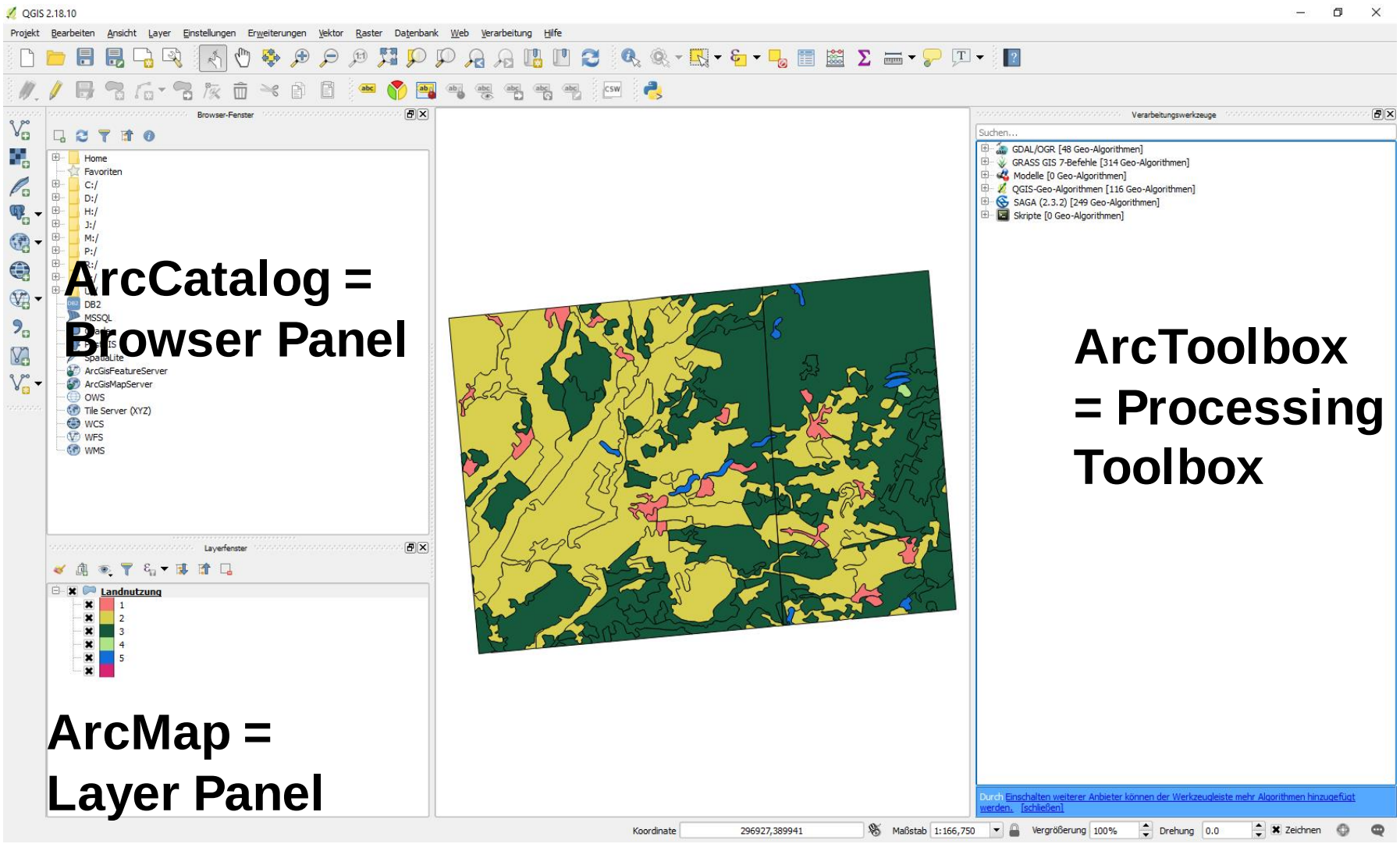
Data Management

ArcToolbox

Analysis

Data Aquisition

Programmaufbau QGIS

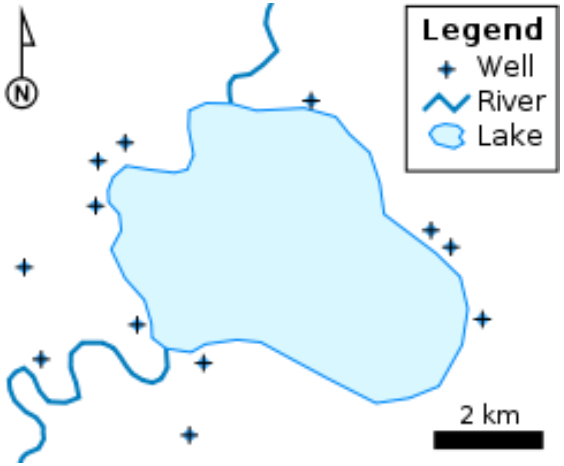


ArcCatalog = Browser Panel

ArcMap = Layer Panel

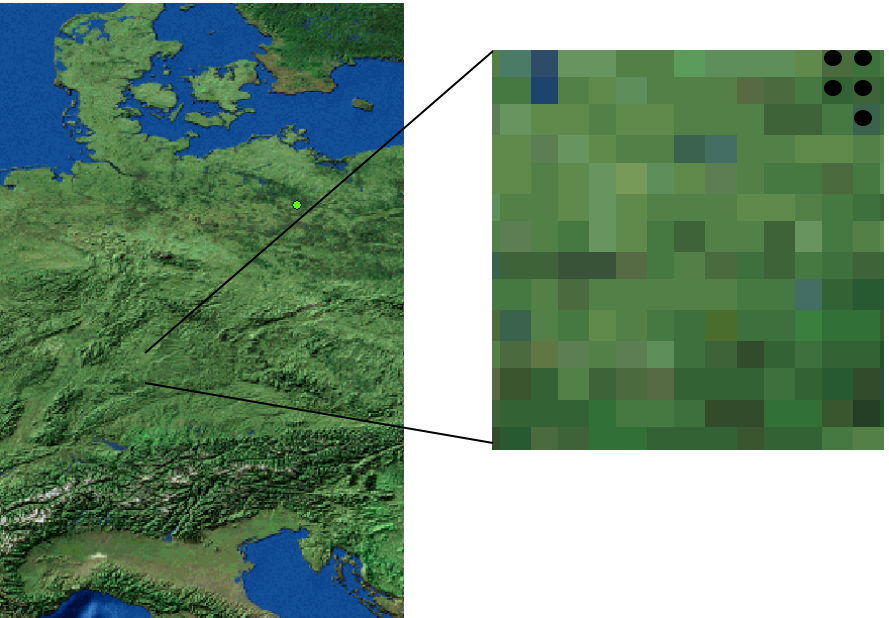
ArcToolbox = Processing Toolbox

Data types – Points Lines Polygons Raster



VECTOR DATA (often shapefiles)

- Point: X/Y coordinates
- Line: Length
- Polygon: area/ perimeter (closed, ‚topology‘)



RASTER DATA

- ‚Coded Pixels‘, like points
- GeoTiffs, ascii, grid, gpkg…

ESRI shapefiles

Shapefiles are a data format developed by ESRI used to hold information on spatial objects. They are pretty ubiquitous and can be used by a lot of GIS packages. Shapefiles can hold polygon, line or point data. Despite the name, a shapefile consists of a few different files:

Mandatory files:

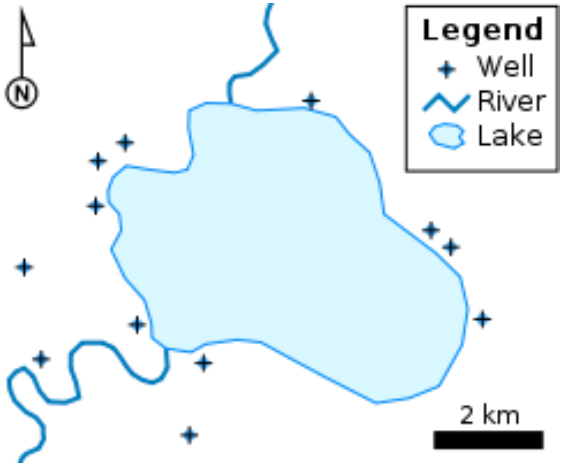
- .shp = The main file containing the geometry data
- .shx = An index file
- .dbf = An attribute file holding information on each object

Common additional files:

- .prj = A file containing information on the Coordinate Reference system
- .shp.xml = a file containing object metadata, citations for data, etc.

From: https://ourcodingclub.github.io/tutorials/maps/

Vector Data format: Sbn,... shx, shp



VECTOR DATA

- Point: X/Y coordinates
- Line: Length
- Polygon: area/ perimeter (closed, ‚topology‘)

-> single vector elements are called *features*

ID	X	Y
1	2	3
1	3	5
1	4	4
1	3	2
1	2	3
2	7	1
2	8	2
2	10	1
2	7	1

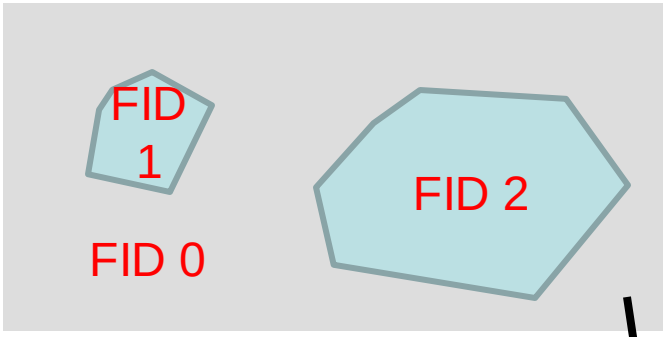
Attributes of test_clip_clc

FID	Shape *	AREA	PERIMETER	CLCGES_BES	CLCGES_B_1	NS	BRD
0	Polygon	130539000	211140	23043	23042	312	1
1	Polygon	33397200	79888.102	23163	23162	312	1
2	Polygon	5757770	18361.199	23196	23195	311	1
3	Polygon	4211380	17164.9	23239	23238	231	1
4	Polygon	11562800	44001.699	23265	23264	211	1

= feature: FID stands for feature-identity

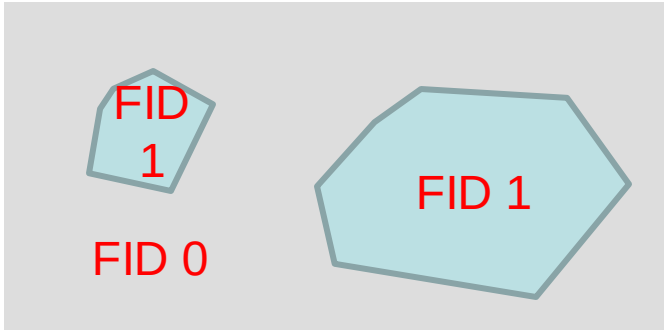
File containing spatial geometry

Vector Data format: Sbn,... shx, shp



‚Union' creates

a multipolygon



e.g. a shapefile with lakes

ID	X	Y
1	2	3
1	3	5
1	4	4
1	3	2
1	2	3
2	7	1
2	8	2
2	10	1
2	7	1

Never change the FID by hand!

Attributes of test_clip_clc

FID	Shape *	AREA	PERIMETER	CLCGES_BES	CLCGES_B_1	NS	BRD
0	Polygon	130539000	211140	23043	23042	312	1
1	Polygon	33397200	79888.102	23163	23162	312	1
2	Polygon	5757770	18361.199	23196	23195	311	1
3	Polygon	4211380	17164.9	23239	23238	231	1
4	Polygon	11562800	44001.699	23265	23264	211	1

In R: *simple feature* vs *spatial polygon df*

```
> test2_sf
Simple feature collection with 158 features and 7 fields
geometry type:  MULTIPOLYGON
dimension:      XY
bbox:           xmin: 108.5986 ymin: -4.943054 xmax: 119.2697 ymax: 7.380556
epsg (SRID):    NA
proj4string:    NA
First 10 features:
   ID_0 ISO    NAME_0 NAME_1      NAME_2 Shape_Leng Shape_Area                       geometry
0   158 MYS Malaysia  Sabah Lahad Datu   6.7475367 0.62106012 MULTIPOLYGON (((118.2299 4....
1   158 MYS Malaysia  Sabah       Papar   1.6727672 0.10065591 MULTIPOLYGON (((116.0144 5....
2   158 MYS Malaysia  Sabah   Penampang   0.9548857 0.03951545 MULTIPOLYGON (((116.0477 5....
3   158 MYS Malaysia  Sabah  Pensiangan   4.0257859 0.49725547 MULTIPOLYGON (((116.5357 5....
4   158 MYS Malaysia  Sabah       Pitas   2.9223456 0.11955352 MULTIPOLYGON (((117.2762 6....
5   158 MYS Malaysia  Sabah       Ranau   2.4815704 0.24027591 MULTIPOLYGON (((116.9182 6....
6   158 MYS Malaysia  Sabah    Sandakan   7.1555231 0.18354676 MULTIPOLYGON (((117.9468 5....
7   158 MYS Malaysia  Sabah    Semporna   6.0198462 0.09494946 MULTIPOLYGON (((118.6294 4....
8   158 MYS Malaysia  Sabah    Sipitang   2.9190984 0.22604467 MULTIPOLYGON (((115.7132 5....
9   158 MYS Malaysia  Sabah    Tambunan   1.5307563 0.12152936 MULTIPOLYGON (((116.6079 5....
```

Closer to ESRI shapefile attribute table

{sf}

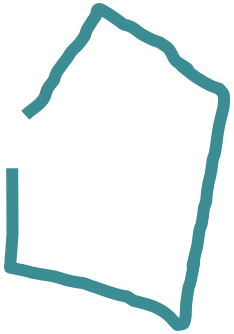
= saved in the geometry as list (POLYGON)
Or as list in a list (MULTIPOLYGON)

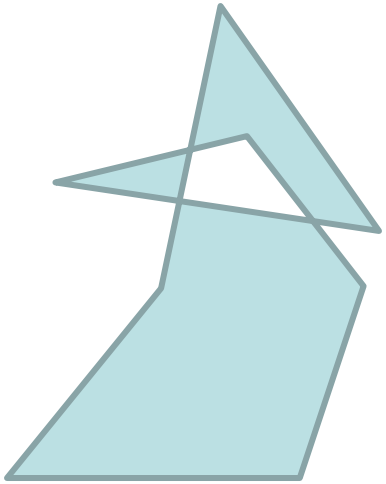
```
class       : SpatialPolygonsDataFrame
features    : 158
extent      : 108.5986, 119.2697, -4.943054, 7.380556  (xmin, xmax
coord. ref. : NA
variables   : 7
names       : ID_0, ISO,    NAME_0, NAME_1, NAME_2, Shape_Leng,   Sh
min values  :   34, BRN,    Brunei, Belait,    Amo,  0.2036641, 0.0
max values  :  158, MYS, Malaysia, Tutong,  Ukong, 30.6175230, 3.1
> |
```

{sp}

Problems with the topology

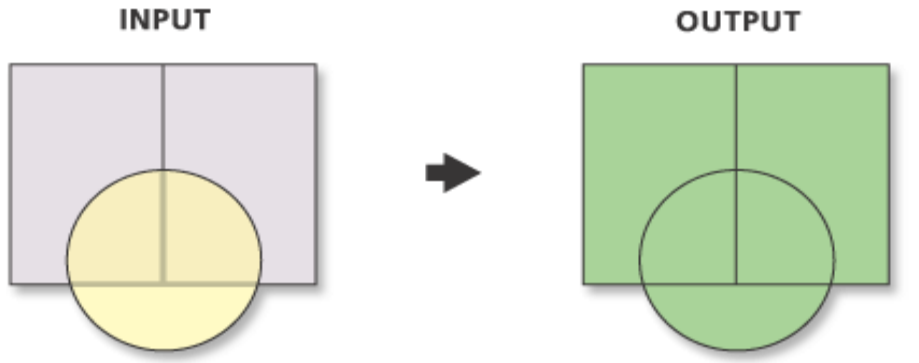
- Geometry not closed or not clean





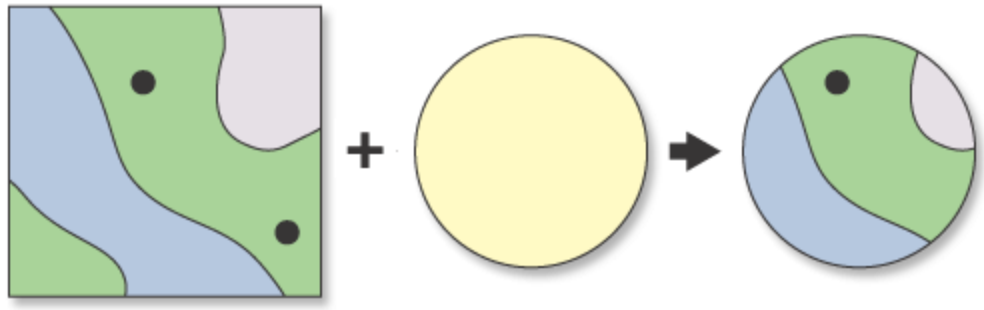
Handling geometry issues:
package ***sf several functions,*** e.g., st_make_valid()
https://r-spatial.org/r/2017/03/19/invalid.html
package ***cleangeo***
https://cran.r-project.org/web/packages/cleangeo/vignettes/quickstart.html

Toolbox: Example of analyses with polygons (shapefiles)



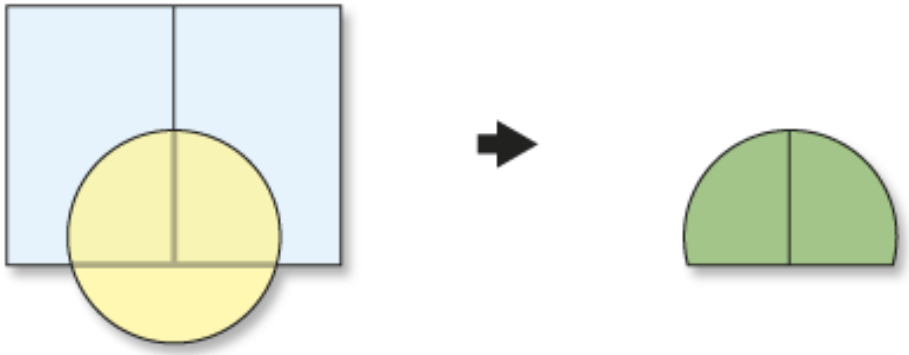
UNION

In R: gUnion()
st_union()



CLIP

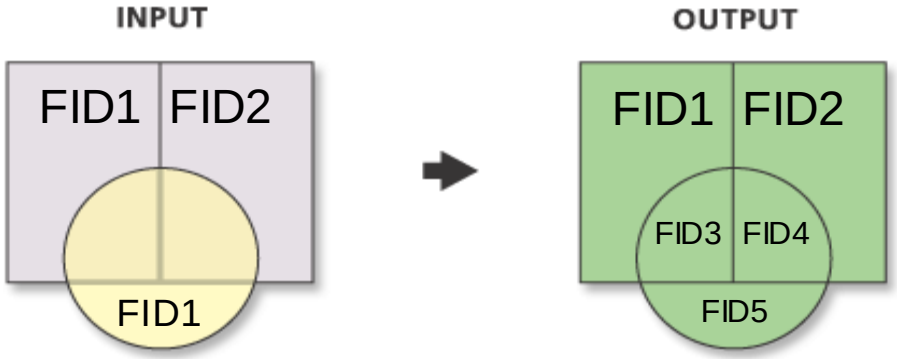
In R: crop()
extract()
gIntersection()
st_Intersection()



INTERSECT

In R:
gIntersection()
st_intersection()

OBS!

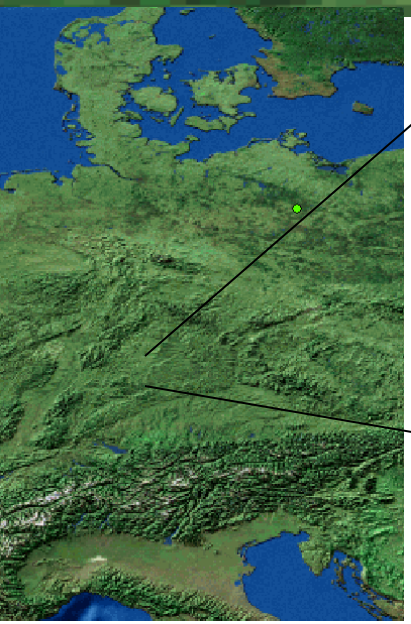


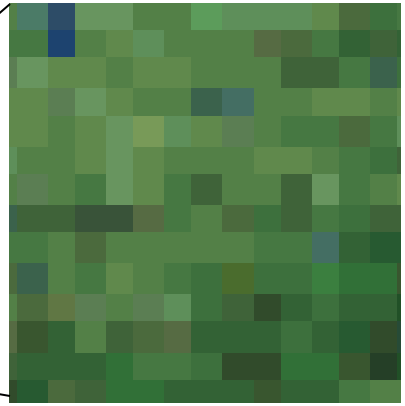
After spatial operation, the FID changes! And the geometry!

And with that also the spatial calculations!

Remember to update/ recalculate the column with the area size/ perimeter/ distances after spatial operations!

Raster format – fast and efficient (Modeling)





xllcorner/
yllcorner

- Best for continuous data
- Ascii or GeoTIFF as exchange format

bio_asc_22.asc - Editor

Datei Bearbeiten Format Ansicht ?

```
ncols         1423
nrows         1425
xllcorner     108.33412288693
yllcorner     -4.3740129986359
cellsize      0.0083333338
NODATA_value  -9999
0 0 0 0 400 5600 400 0 0 0 1600 0 0 0 0 1600 4400
6400 1200 0 0 0 0 0 0 0 0 0 0 0 0 0 1200 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 2000 2000 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
1600 2800 1600 0 0 0 0 800 0 0 4400 800 0 2000 400
2800 400 3200 2400 3200 4400 1600 0 0 0 0 0 0 0 0
0 0 0 4800 6000 800 0 1600 2000 800 3600 400 0 0
400 4000 2000 0 0 0 0 0 0 400 5600 1600 0 0 0 0
6400 0 5600 7200 0 0 0 0 0 0 0 0 0 0 0 3600 8400
6800 0 0 0 0 0 0 400 4000 6000 6800 800 3600 0 0 0
0 0 1600 400 0 0 1200 6800 9600 10000 10000 10000
10000 10000 9600 3600 4000 9600 1200 0 400 2400
10000 9600 4400 1200 400 1600 800 0 0 0 0 0 0 0 0
0 0 0 0 0 2000 7600 10000 10000 9600 9600 8000
7600 9200 8000 10000 10000 10000 10000 10000 10000
9200 3200 4400 2000 3200 9600 2400 0 4000 9200
5600 3200 0 0 0 0 0 0 0 0 0 0 -9999 -9999 -9999 -
9999 -9999 -9999 -9999 -9999 -9999 -9999 -9999 -
9999 -9999 -9999 -9999 -9999 -9999 -9999 -9999 -
```

In R – raster (getting outdated) and terra

- Bor_mat: Borneo mean annual temperature raster

```
Bor_mat <- raster(paste(raster_wd,'Borneo_MAT.asc',sep='/'))
Bor_mat
```

```
## class       : RasterLayer
## dimensions  : 1425, 1423, 2027775  (nrow, ncol, ncell)
## resolution  : 0.008333334, 0.008333334  (x, y)
## extent      : 108.3341, 120.1925, -4.374013, 7.500988  (xmin, xmax, ymin, ymax)
## coord. ref. : NA
## data source : C:/Users/kramer/Dropbox/_GeoData_Mix/Geodata/rasterfiles/Borneo_MAT.
asc
## names       : Borneo_MAT
```

(raster) / SpatRaster (terra)

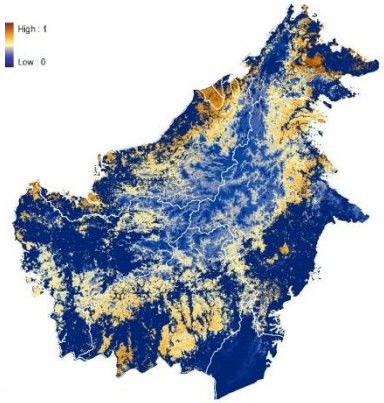
S4-format in raster: slots are accessible via @, i.e.

```
Bor_mat@extent
```

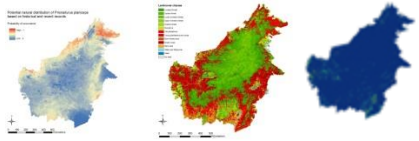
```
## class       : Extent
## xmin        : 108.3341
## xmax        : 120.1925
## ymin        : -4.374013
## ymax        : 7.500988
```

Spatial analyses with rasters

- Manipulation (eg reclassification)
- Algebra (eg calculations based on several maps)



$$HSI = (M * L * H)^{1/3}$$

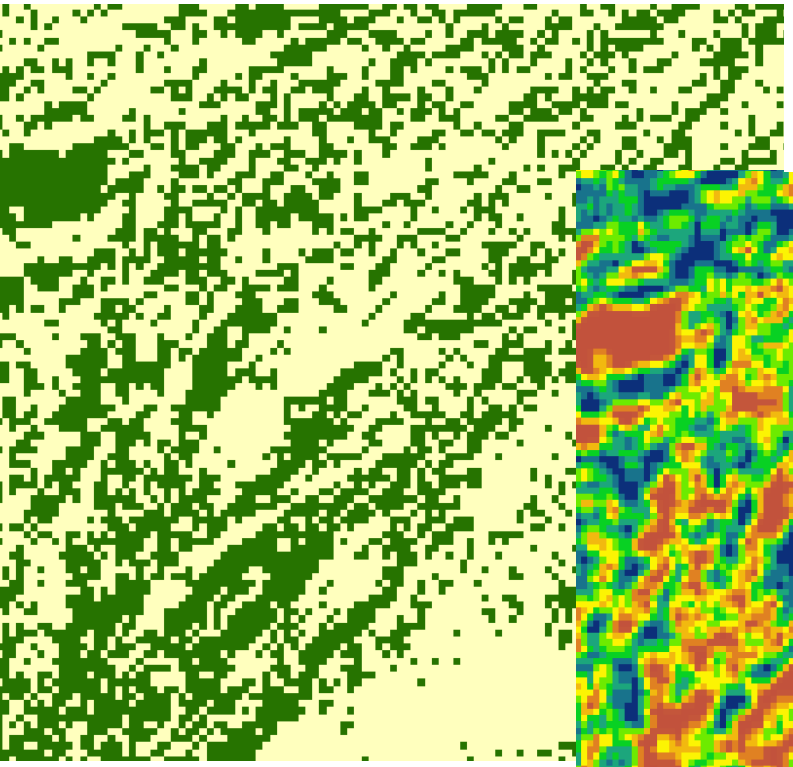


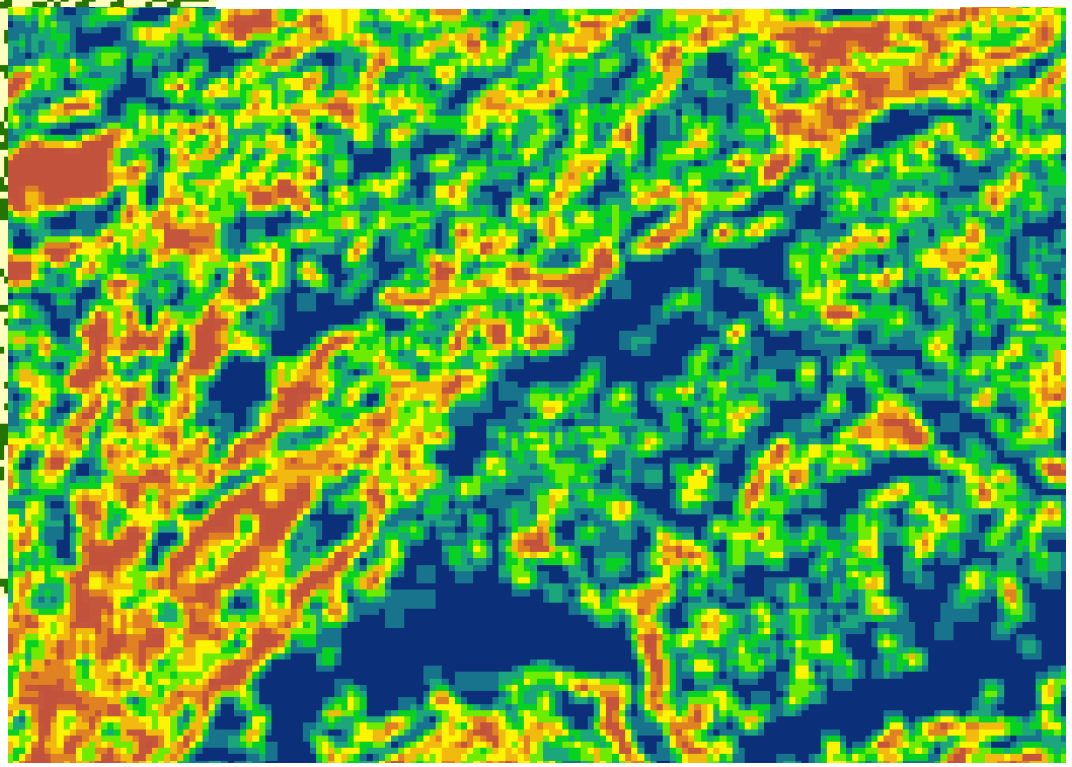
$$= \left(\begin{array}{|c|c|c|} \hline 7 & 10 & 3 \\ \hline 11 & 11 & 8 \\ \hline 8 & 10 & 6 \\ \hline \end{array} * \begin{array}{|c|c|c|} \hline 5 & 7 & 2 \\ \hline 9 & 10 & 4 \\ \hline 8 & 8 & 3 \\ \hline \end{array} * \begin{array}{|c|c|c|} \hline 2 & 3 & 1 \\ \hline 2 & 1 & 4 \\ \hline 0 & 2 & 3 \\ \hline \end{array} \right)^{1/3}$$

- Modelling (NDVI, ruggedness, neighborhood analyses, cost path, watershed,…)

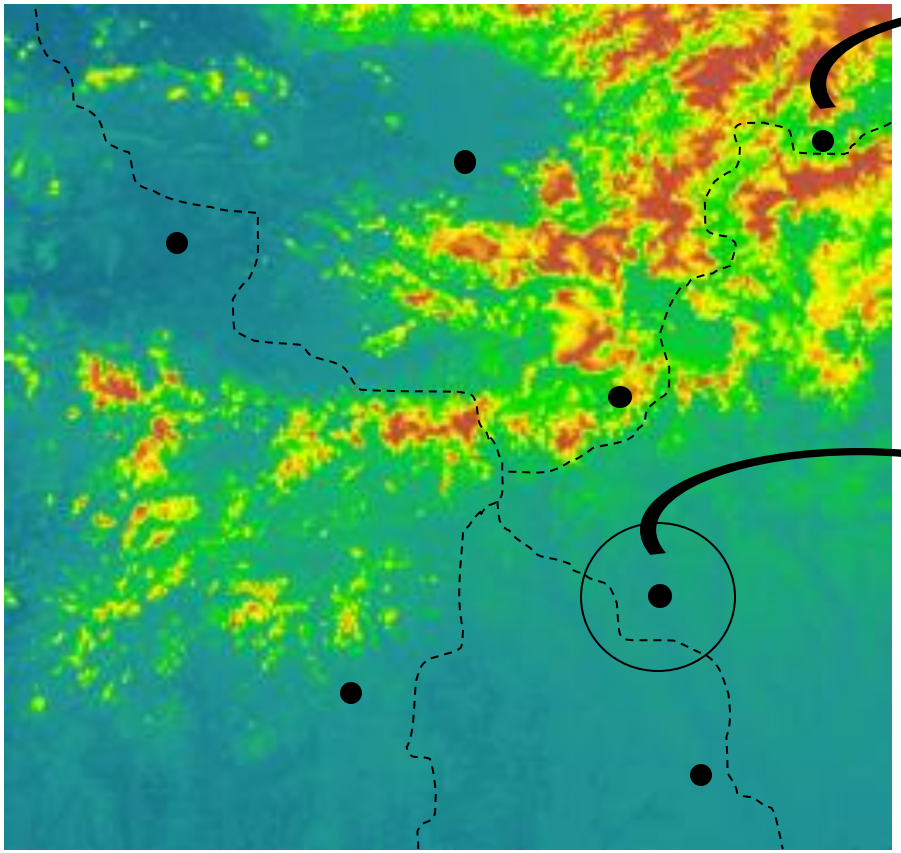
Spatial analyses with rasters II

e.g. Connectivity indices (Focal Statistics):
To describe single point values across larger area





Analysis: Building Databases With Spatial Info



Per Point (Extraction):
- Map value
  (Land use, climate)
- area/ perimeter of patch
- distances to roads…

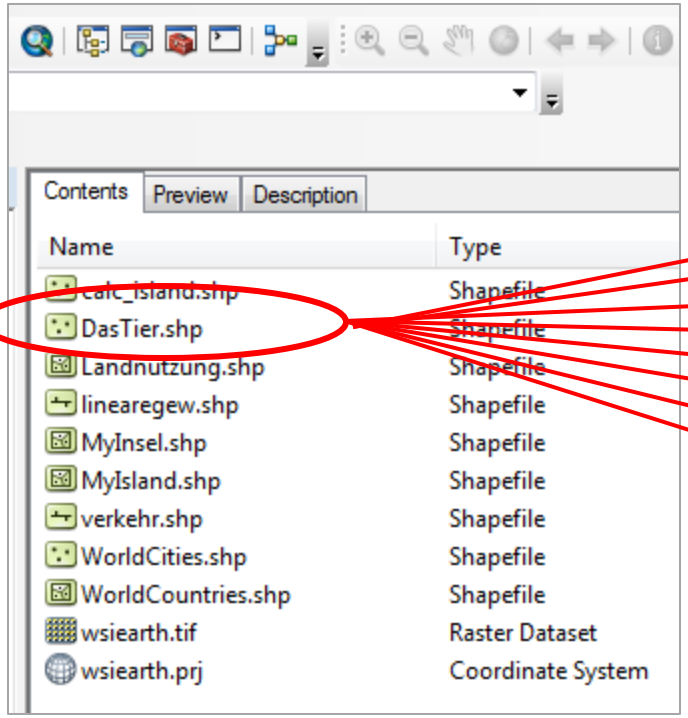
In R:
extract
over
st_join

Per buffer (Clip):
- % land use types in buffer
- road length…

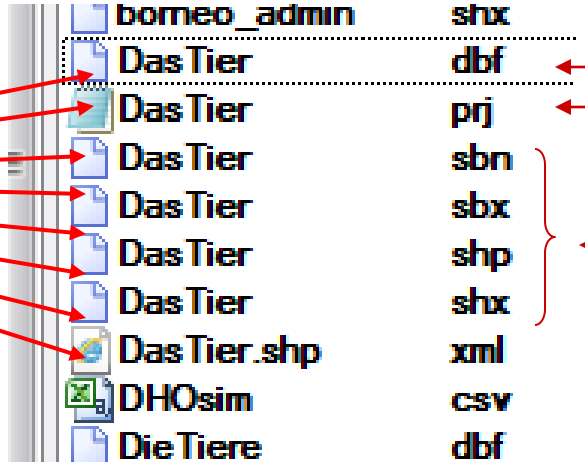
In R:
gBuffer
st_buffer

ArcCatalog – Data Formats

ArcCatalog



‚Normal' Explorer



Factual info

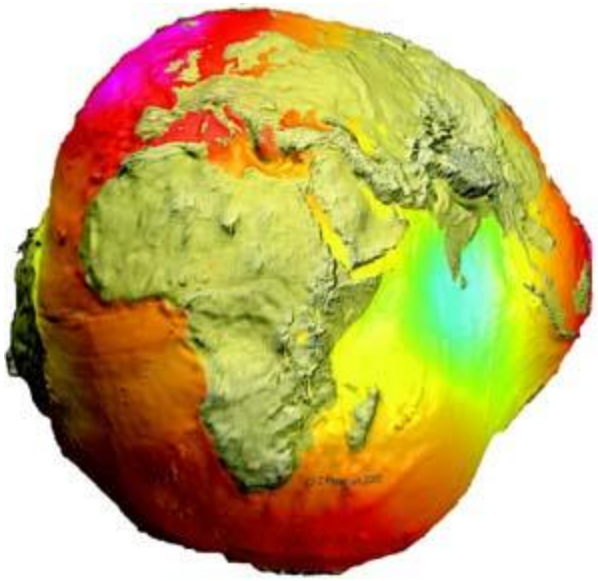
Projection

Geo info

If you copy-paste-rename spatial polygon layers, you need to do it with all files

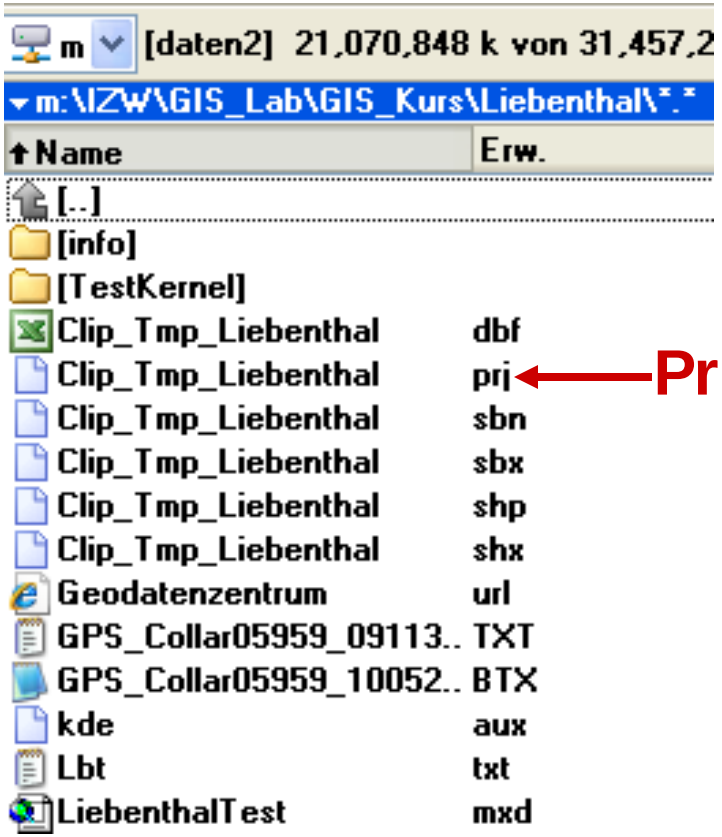
- Copying/renaming of data files and moving of projects only via ArcCatalog!

Coordinate Systems and Projections



How to describe where we are…

‚Normaler' Explorer



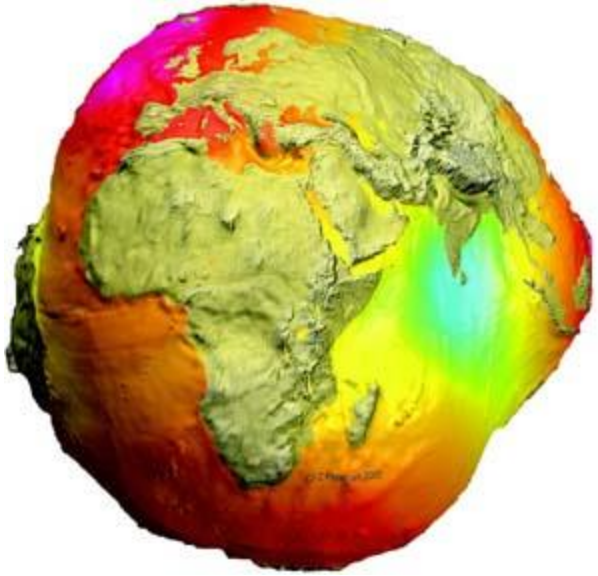
PROJCS["DHDN_3_Degree_Gauss_Zone_4",GEOGCS["GCS_Deutsche_Hauptdreiecksnetz",DATUM["D_Deutsche Hauptdreiecksnetz",SPHEROID["Bessel_1841",6377397.155,299.1528128]],PRIMEM 0.0174532925199433]],PROJECTION["Gauss_Kruger"],PARAMETER["False_Eastin e_Northing",0.0],PARAMETER["Central_Meridian",12.0],PARAMETER["Scale_Factor Origin",0.0],UNIT["Meter",1.0]]

=WKT well-known textfile
-> future use in spatial R

Where are we? – Reference System

Ball (Sphere)?
(Pythagoras
500 BC)

Ellipse?
That‘s better!
(Sir Isaac Newton)

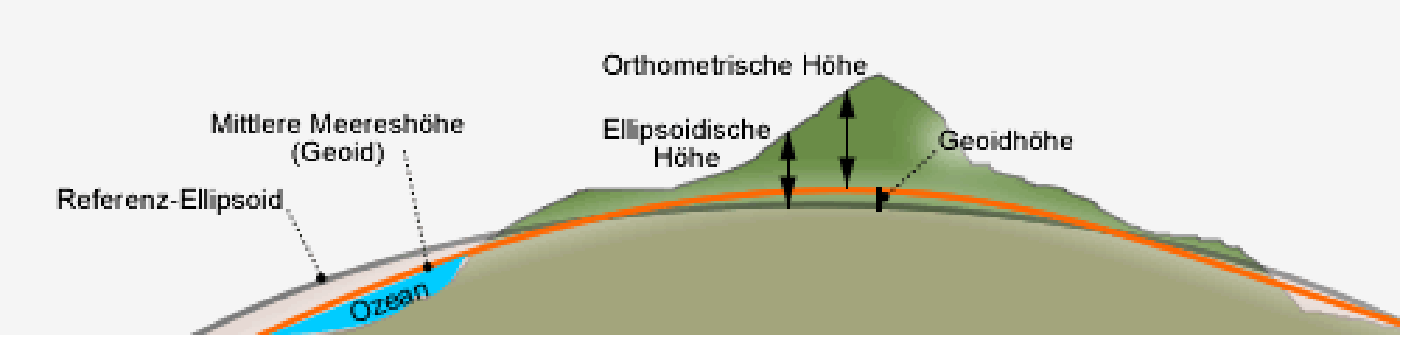


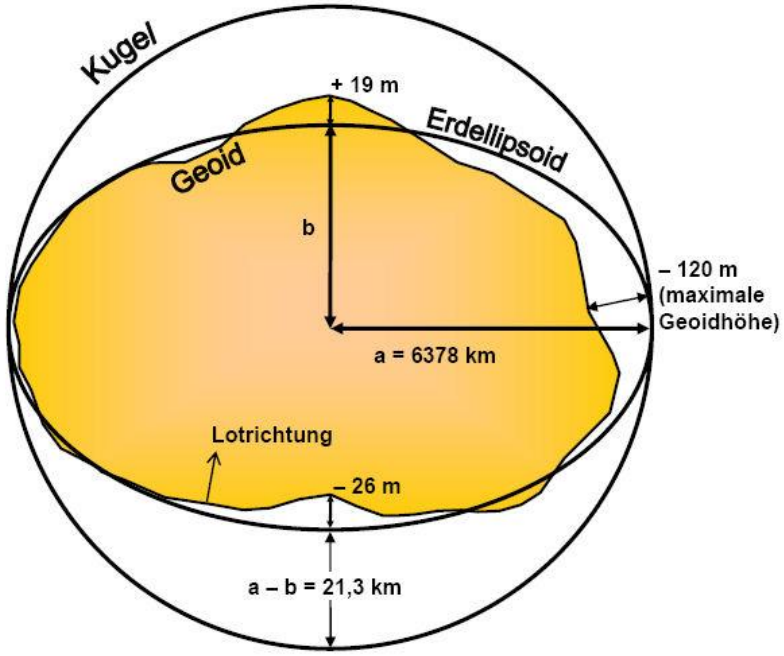
Geoid!
(Gauß)

Colour code: deviation
from reference ellipsoid

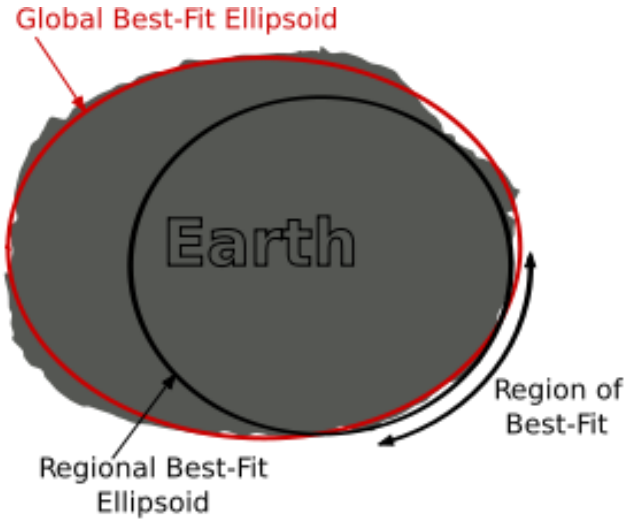
Approximation of a geoid…

Easier to describe ellipsoids mathematically





Geographic coordinate system (GCS)



Usually given in angular units (latitude/ longitude)

Also includes:
Ellipsoid = size and shape of ellipse
Ellipsoids (spheroids) mathematically more easily described

Geodetic datum = position and orientation of reference ellipsoid

Global Datum: WGS84 with GRS80-Ellipsoid ← **GPS !!!**

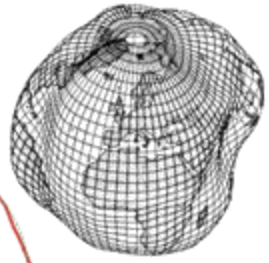
World Geodetic System 1984
With Geodetic Reference System 1980

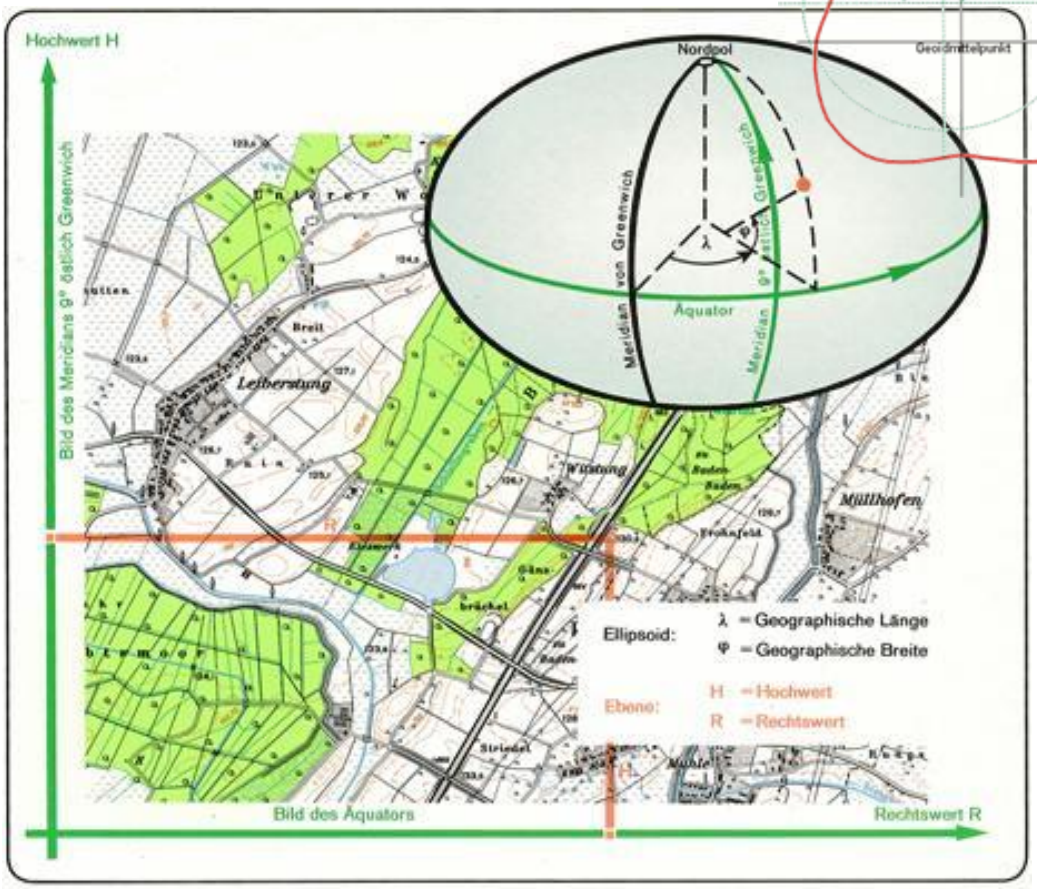
Local Datum: eg
Potsdam (Rauenberg)
with Bessel-Ellipsoid

http://de.wikipedia.org/wiki/Geod%C3%A4tisches_Datum

Projected coordinate systems (PCS)
From 3D to 2D

What if I want distances in metres or area sizes in m² ?

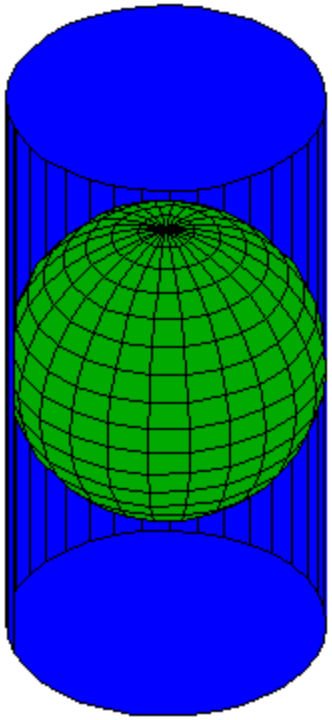




- constant lengths, angles, and areas across the two dimensions
- linear unit of measure

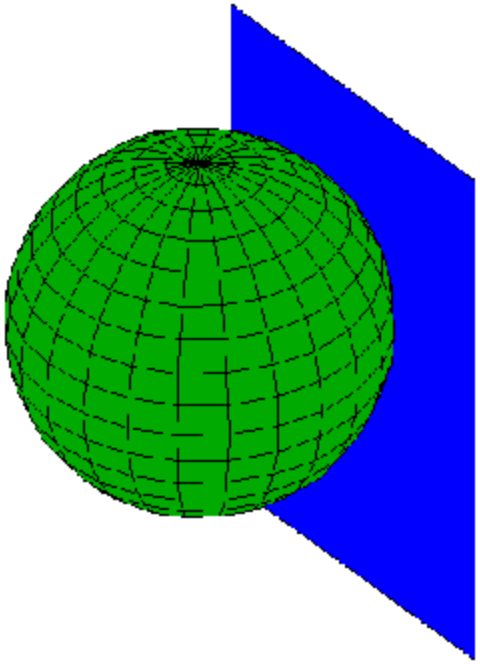
Projections

cylindric



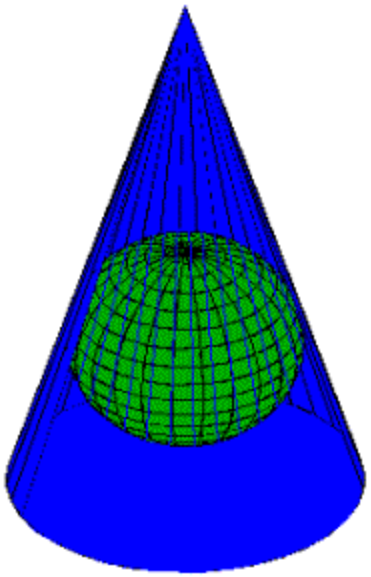
eg UTM

Planar/
azimuthal



eg Lambert

conic

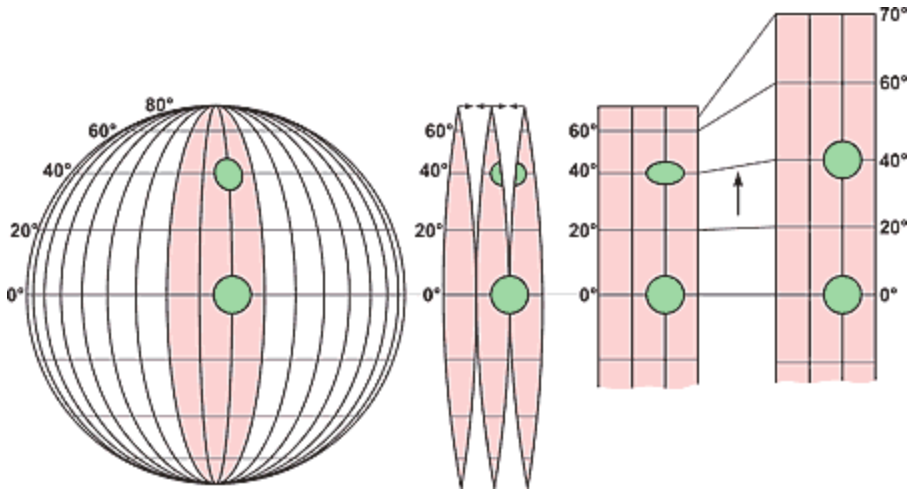


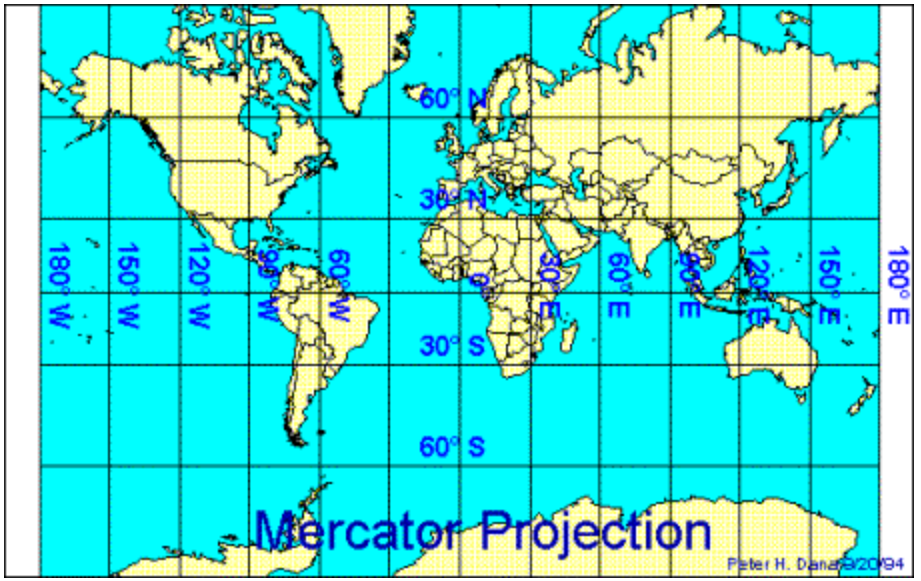
eg Albers

eg Mercator projections (conformal)
Gauss-Krüger (in 3°/ 6° stripes), UTM (equal area-projection)

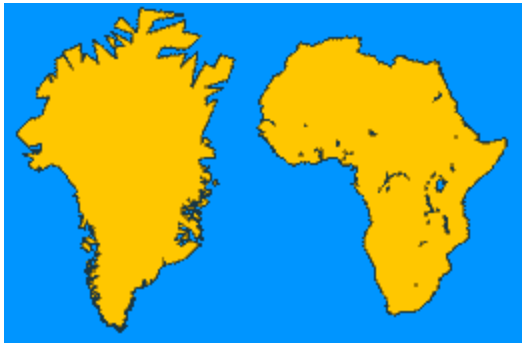
Quelle: Peter H. Dana 1994, Uni Siegen
http://resources.esri.com/help/9.3/arcgisengine/dotnet/89b720a5-7339-44b0-8b58-0f5bf2843393.htm

Example: Cylindrical Projection





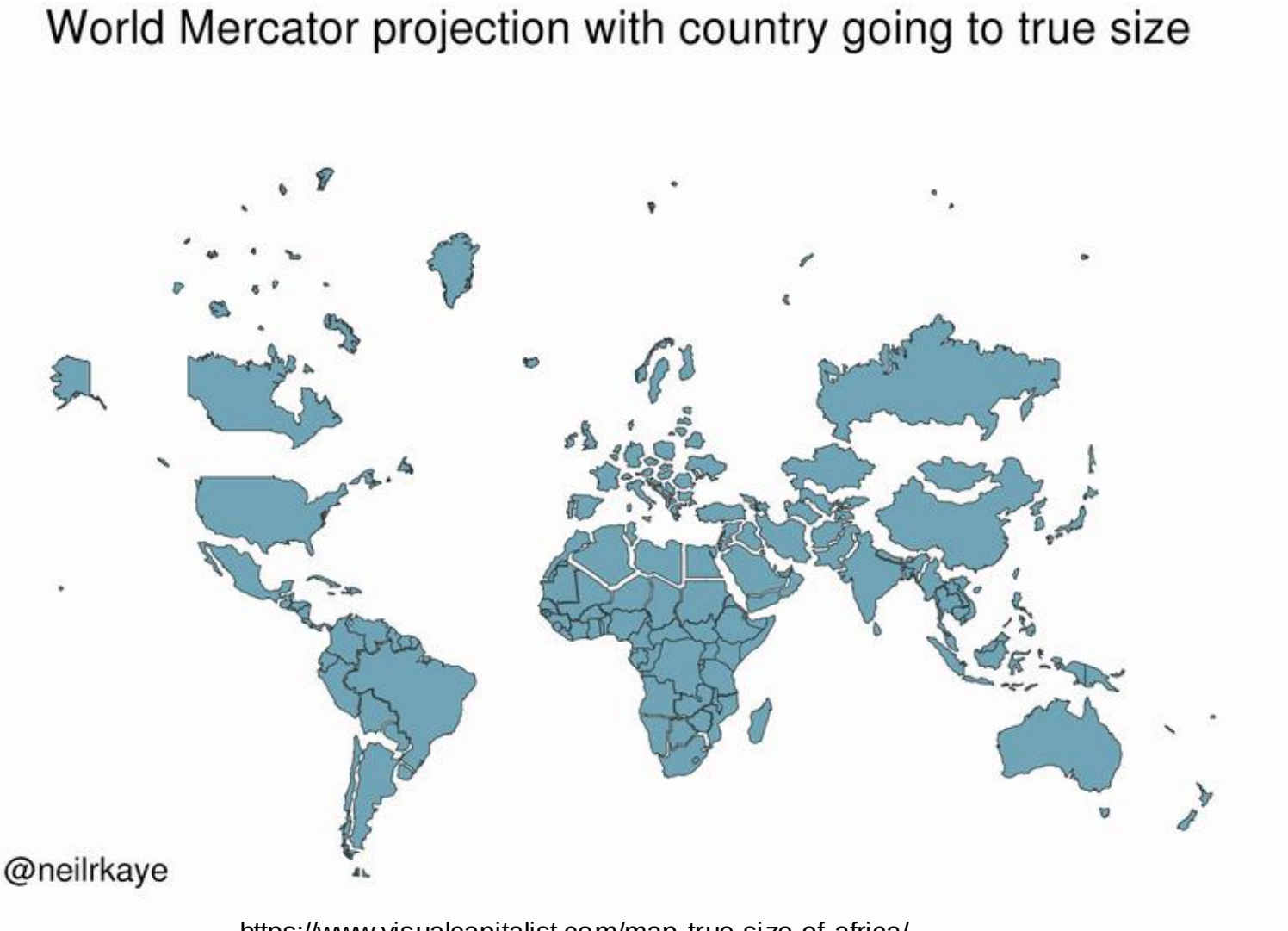
geographic



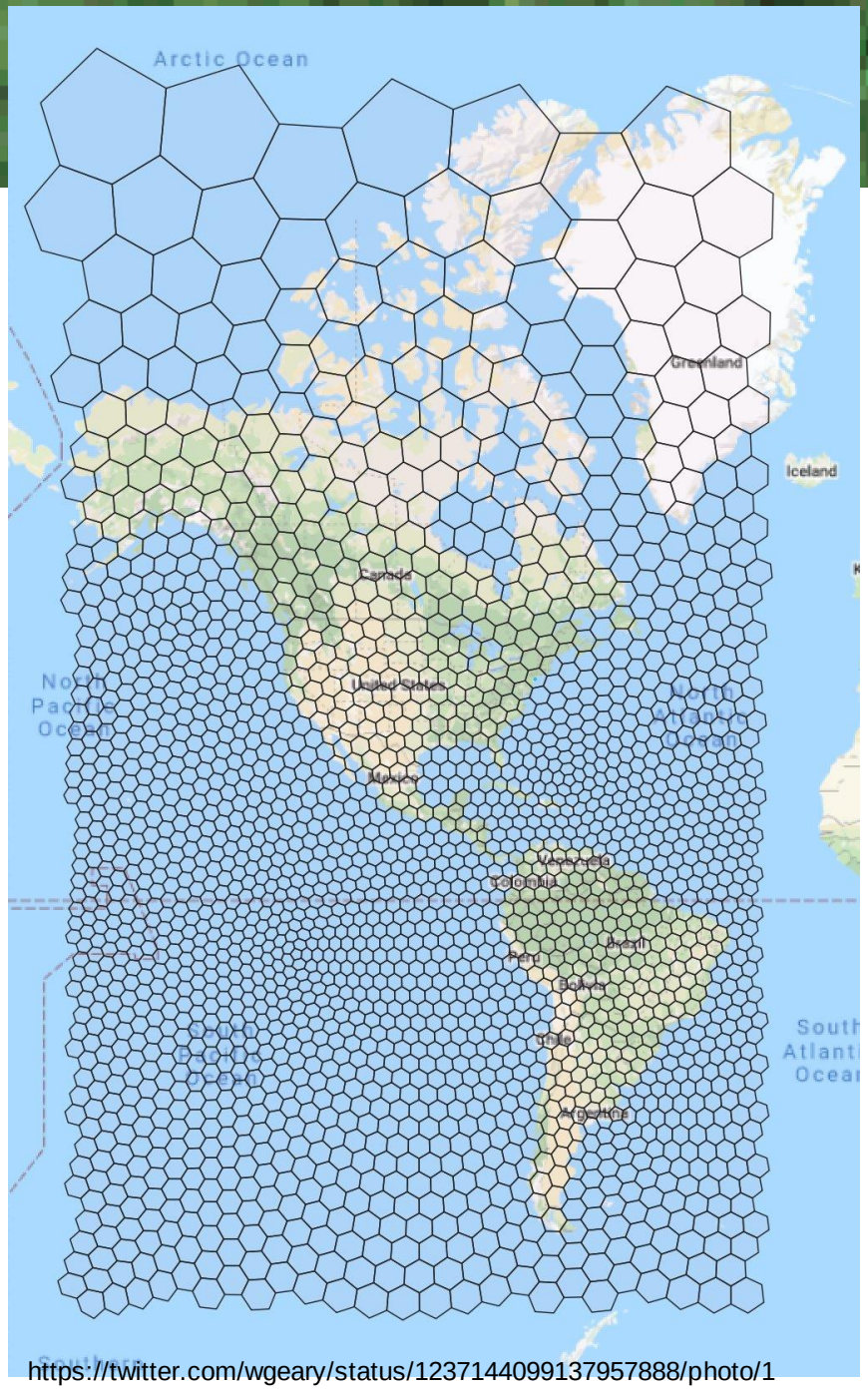
projected

Comparison of size between Greenland and Africa in the Mercator projection which is conformal but not an equal-area projection (left) and in the equal-area Lambert projection (right)

- https://twitter.com/i/status/1237163708285104129



https://www.visualcapitalist.com/map-true-size-of-africa/



https://twitter.com/wgeary/status/1237144099137957888/photo/1

https://geoawesomeness.com/best-map-projection/

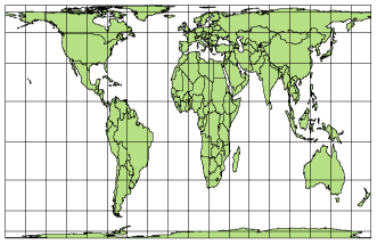
http://de.wikipedia.org/wiki/Kartennetzentwurf

Mollweide-Projektion

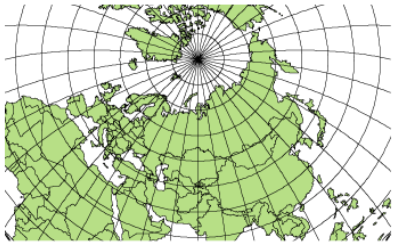
Mercator-Projektion

Zylinderprojektion nach Miller

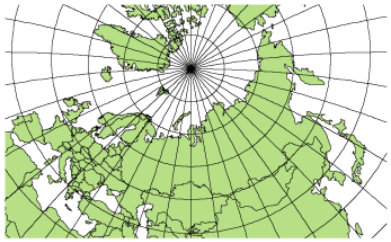
Hammer-Aitoff-Projektion



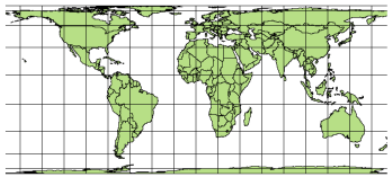
Peters-Projektion



Längentreue Azimuthalprojektion



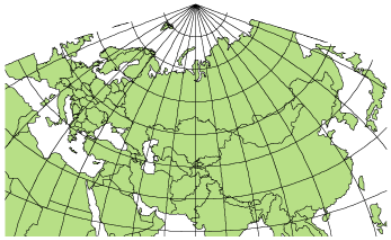
Stereographische Projektion



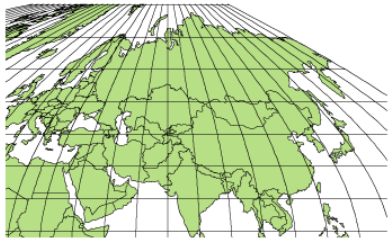
Behrmann-Projektion

Senkrechte Umgebungsperspektive

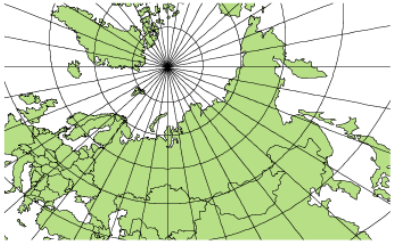
Robinson-Projektion



Hotine Oblique Mercator-Projektion

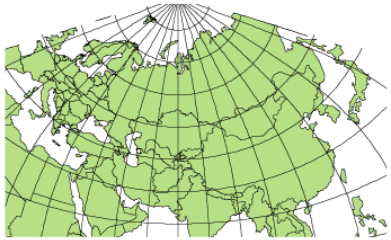


Sinusoidale Projektion

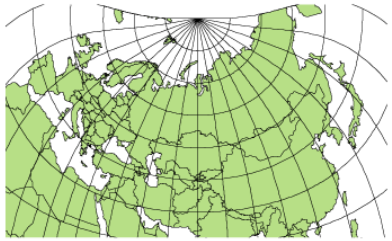


Gnomonische Projektion

Flächentreue Kegelprojektion



Transverse Mercator-Projektion



Cassini-Soldner-Projektion

Summary

- Ellipsoid: describes form and size of ellipse (Bessel, Clarke, Krassowski…)
- Date/ datum: origin and orientation of the ellipsoid
- Projection: mapping rule concerning depiction of sphere/ellipsoid/geoid in 2D (plane)

Projection-file (.prj) in ArcGIS describes this to be able to display maps of different projections and coordinates on top of each other.

PROJCS["DHDN_3_Degree_Gauss_Zone_4",GEOGCS["GCS_Deutsche_Hauptdreiecksnetz",DATUM["D_Deutsche_Hauptdreiecksnetz",SPHEROID["Bessel_1841",6377397.155,299.1528128]],PRIMEM["Greenwich",0.0],UNIT["Degree",0.0174532925199433]],PROJECTION["Gauss_Kruger"],PARAMETER["False_Easting",4500000.0],PARAMETER["False_Northing",0.0],PARAMETER["Central_Meridian",12.0],PARAMETER["Scale_Factor",1.0],PARAMETER["Latitude_Of_Origin",0.0],UNIT["Meter",1.0]]

In R – crs – coordinate reference system

- EPSG stands for **European Petroleum Survey Group**. They publish a database of coordinate system information plus some very good related documents on map projections and datums.

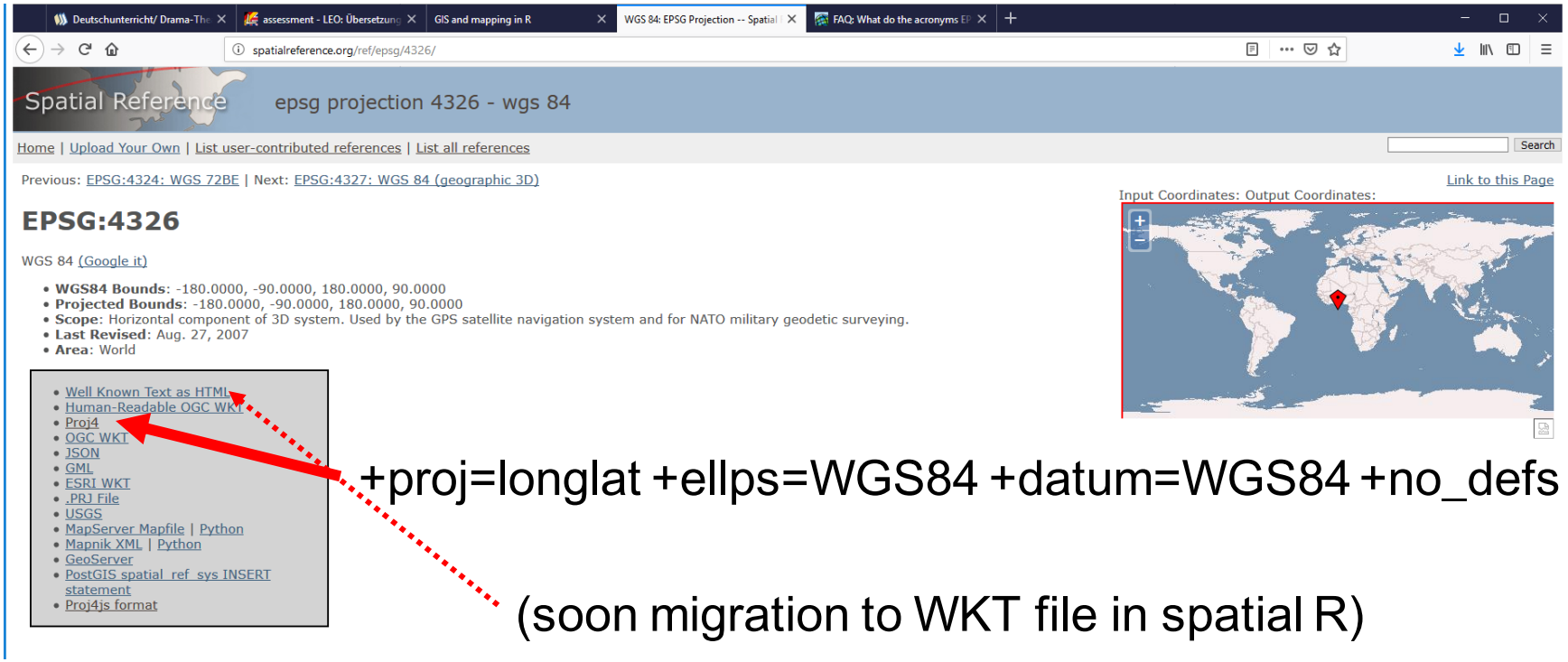
- https://support.esri.com/en/technical-article/000002814

- http://spatialreference.org/

In R:
{raster}:
raster@crs <- CRS("+init=epsg:4326")

{terra}
crs(ras_bio_asc_01) <- "+init=epsg:4326"

EPSG code WGS84: 4326



In R:

{raster}:

```
raster@crs <- CRS("+init=epsg:4326")
```

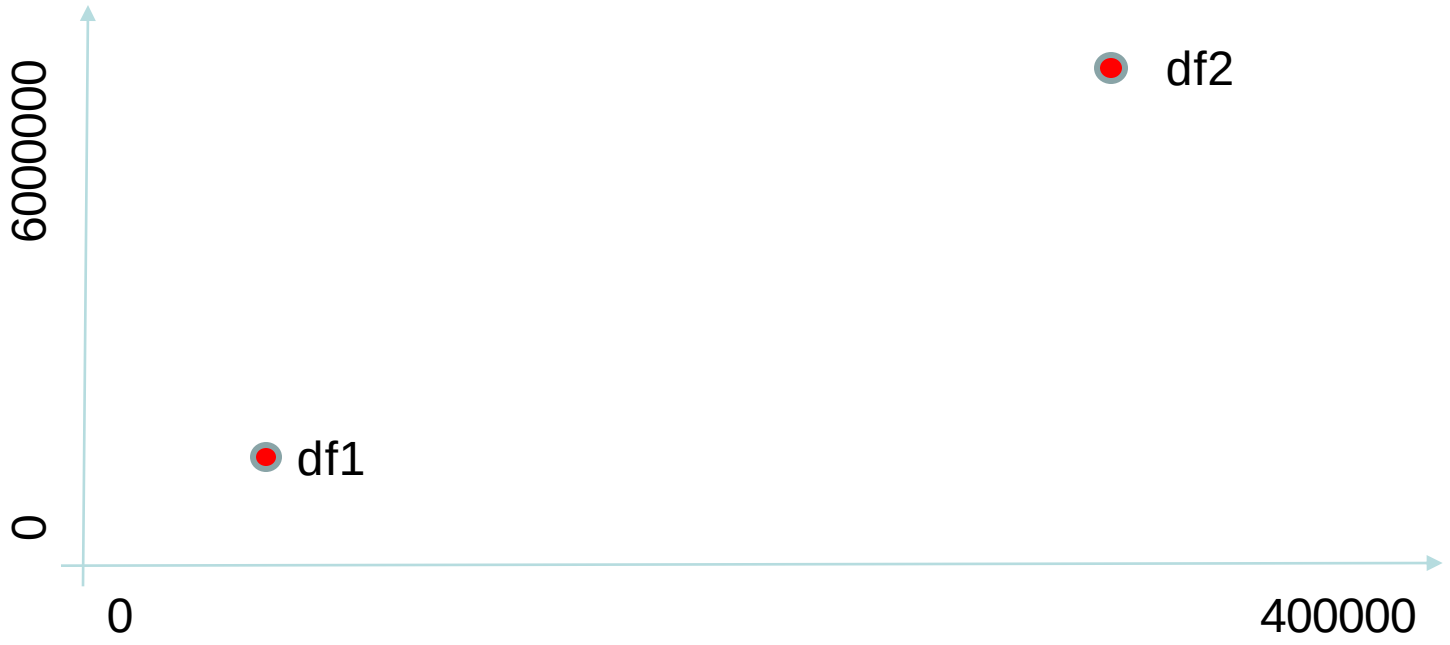
{terra}

```
crs(ras_bio_asc_01) <- "+init=epsg:4326"
```

Common CRS errors – assignment & projection/ transformation

Data.frame 1	Longitude	Latitude
	13.0	52.0
Data.frame 2	X	Y
	390244.32	5823218.29

ASSIGN the CRS: st_crs()

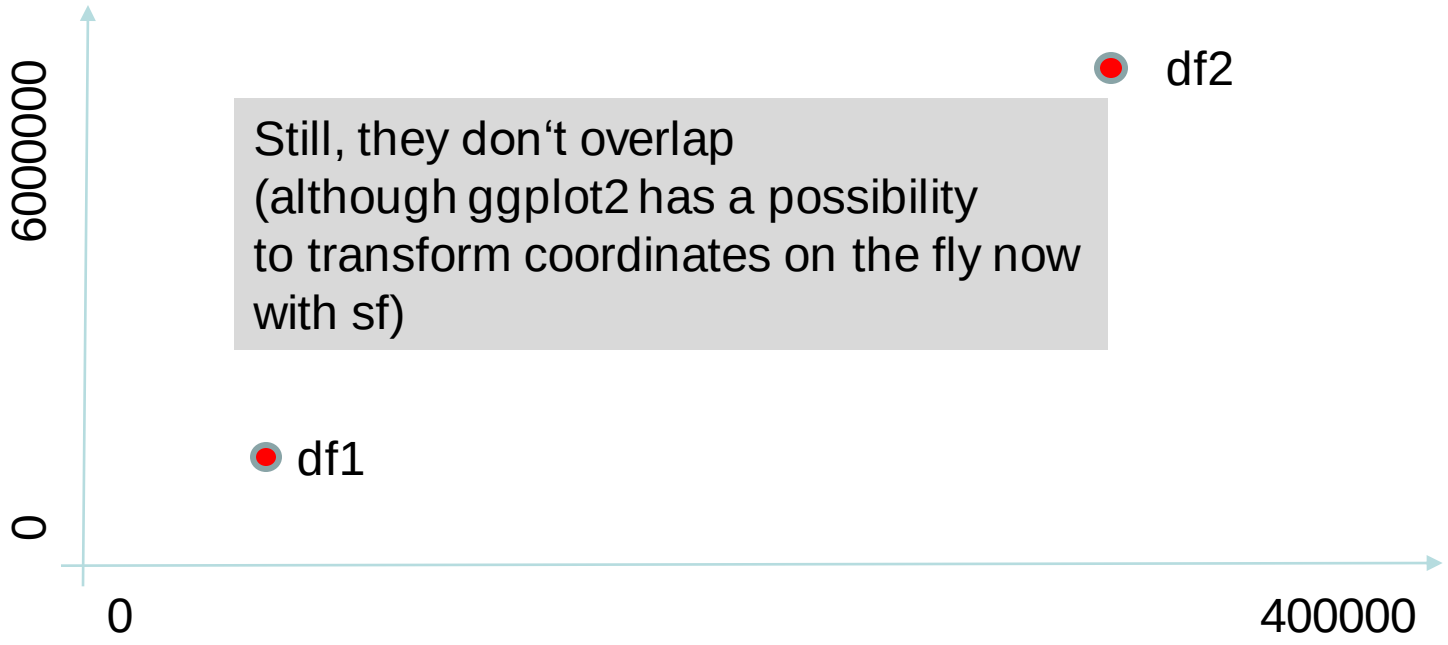


Common CRS errors – assignment & projection/ transformation

Data.frame 1	Longitude 13.0	Latitude 52.0
Data.frame 2	X 390244.32	Y 5823218.29

WGS84 – angular (°)

LAEA 32633 – planar (m)

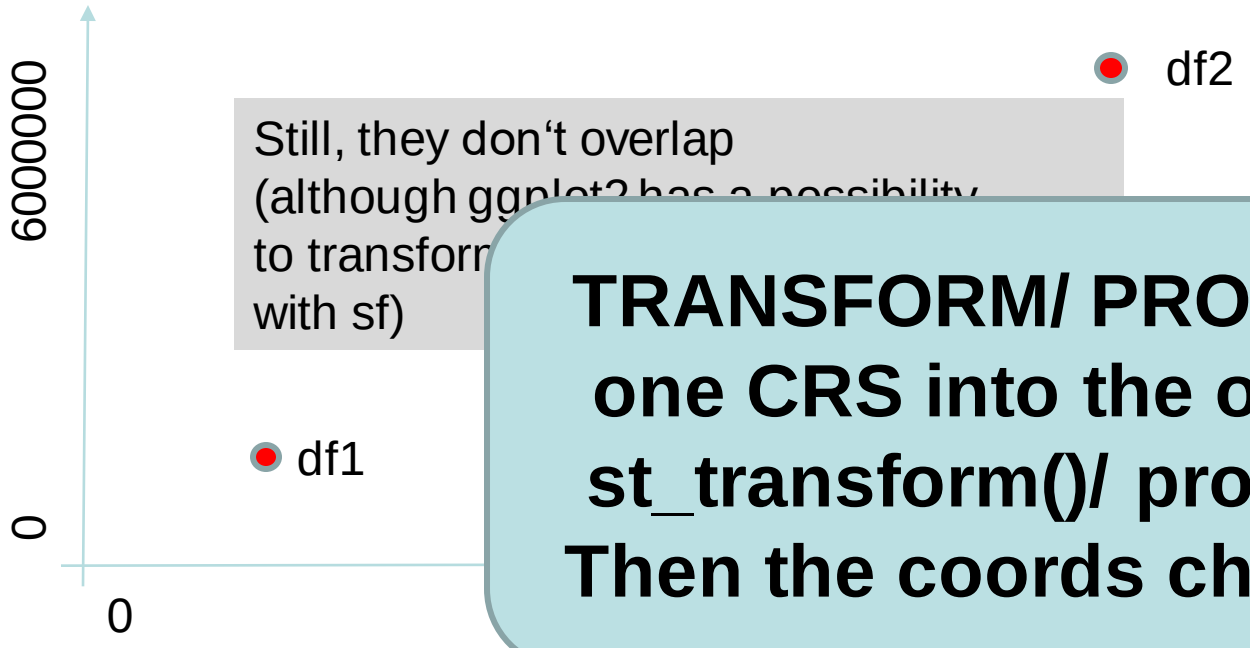


Common CRS errors – assignment & projection/ transformation

Data.frame 1	Longitude 13.0	Latitude 52.0
Data.frame 2	X 390244.32	Y 5823218.29

WGS84 – angular (°)

LAEA 32633 – planar (m)



TRANSFORM/ PROJECT one CRS into the other: st_transform()/ project() Then the coords change!

Visualisation in R

- https://drive.google.com/file/d/1de3dTrAj90e2R9zp9gYPy2kdCjbzdvvK/view
- Slide by Giulia Ruggeri

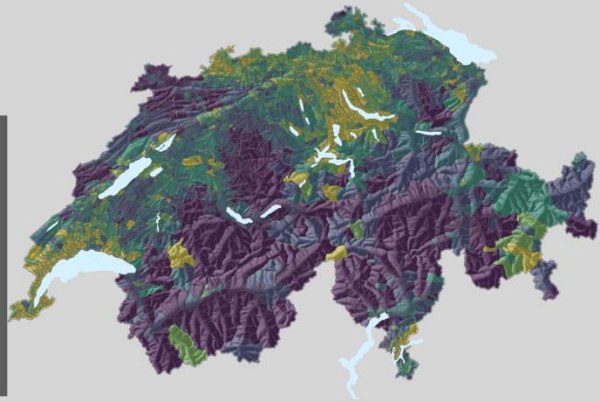
{ggplot2} : geom_sf()


```
# plot the data

> ggplot() +
      geom_sf(data = country) +
      geom_sf(data = lakes, fill = "#D6F1FF") +
      geom_sf(data = districts,
                  aes(fill = income_categories))
```

{ggplot2} : geom_raster( ) + geom_sf( )

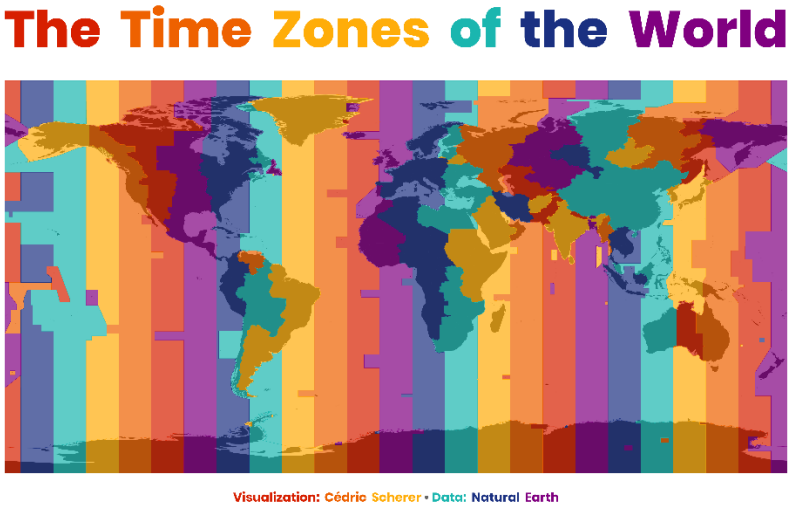
```
> ggplot() +
    geom_raster(data = relief,
                aes(x = x,
                    y = y,
                    alpha = relief)) +
     geom_sf(data = country) +
     geom_sf(data = lakes, fill = "#D6F1FF")+
     geom_sf(data = districts,
                aes(fill = income_categories))
```

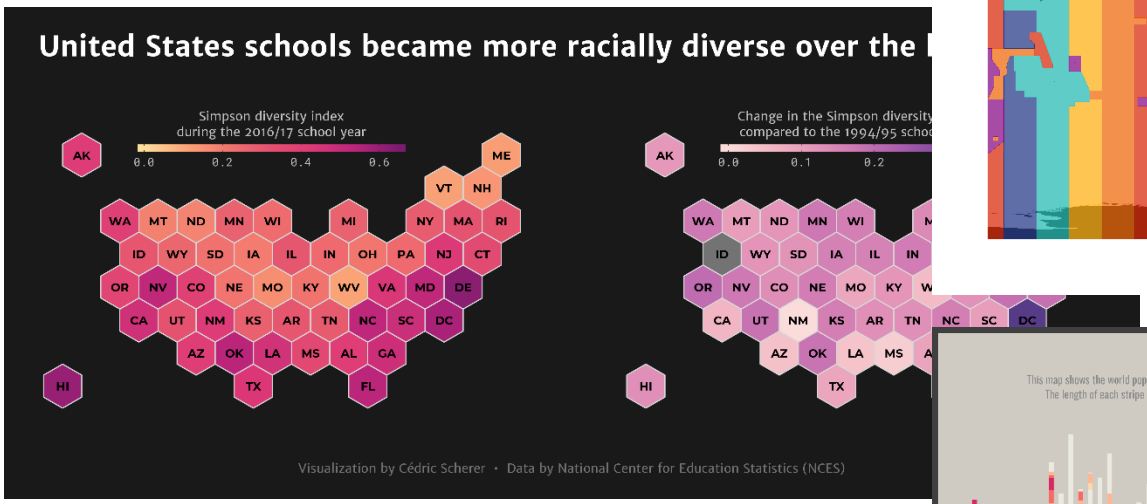


To be done in the hands-on R course.

Viz: Get inspired by Cédric & ggplot2

https://www.data-vizard.com/





Summary: The main commands

Vector data

Raster data

sf	**sp**	**terra**	**raster**
st_read / st_as_sf	readOGR()	rast()	raster()
st_crs()	proj4string()	crs()	crs()
st_transform()	spTransform()	project()	projectRaster()
st_write()	writeOGR()	writeRaster()	writeRaster()

as(x,‘Spatial‘)

as(x,‘sf‘)

Data manipulation similar to base R / (data.frame, matrix, list, S4)

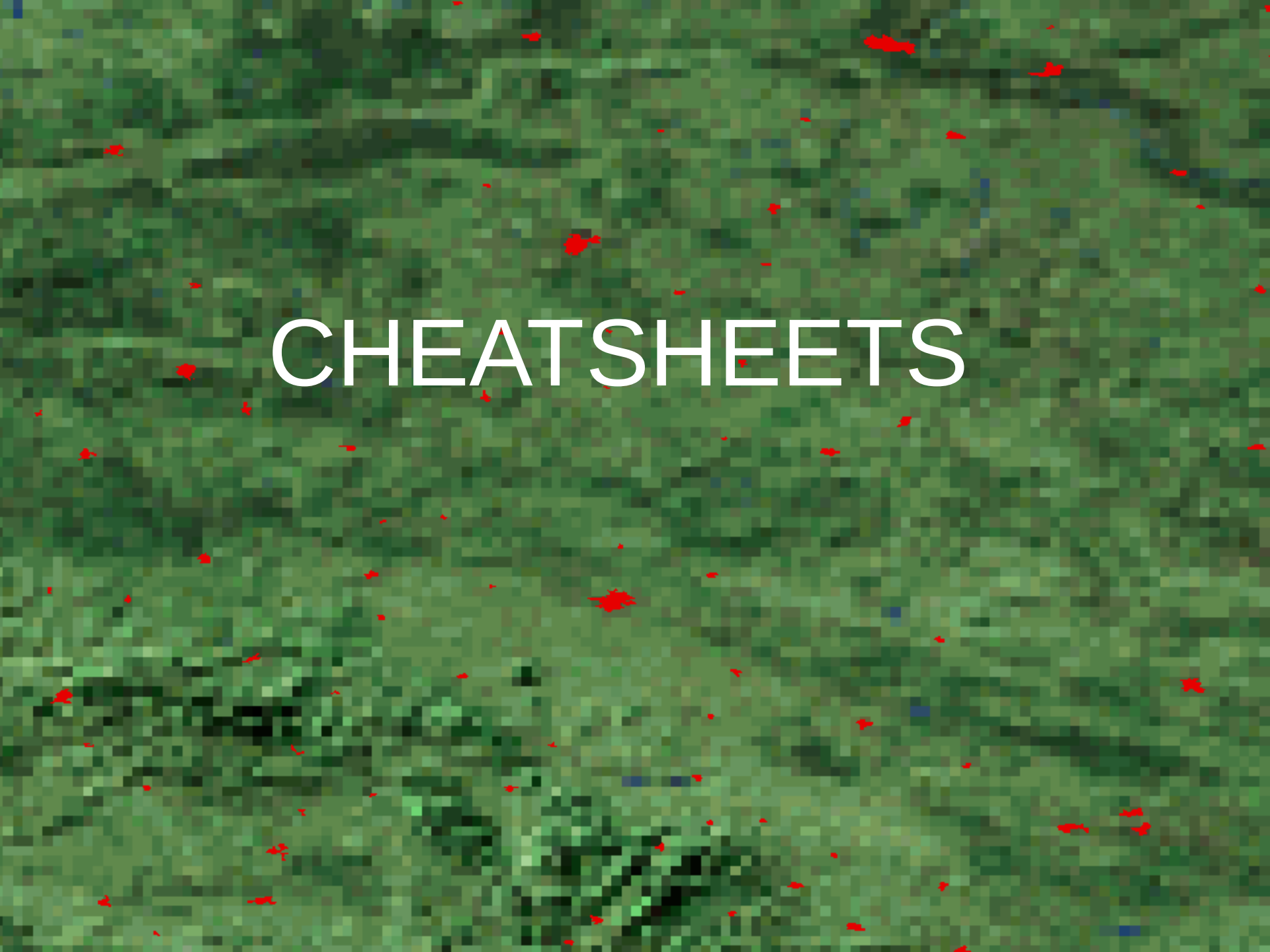
Summary: The main commands

Vector data		Raster data	
sf	**sp**	**terra**	**raster**
st_read / st_as_sf st_crs() st_transform()	readOGR() proj4string() spTransform()	rast() crs() project()	raster() crs() projectRaster()

TAKE HOME MESSAGE: spatial R is the same as handling basic data types (data.frame, matrix, list) – just that you need to understand the added spatial component handling (*geometry*)

Use the course scripts for quick lookup (like a vocabulary book….) and make your own notes


Practical things…


CHEATSHEETS

Base R
Cheat Sheet

Getting Help

Accessing the help files

?mean
Get help of a particular function.
help.search('weighted mean')
Search the help files for a word or phrase.
help(package = 'dplyr')
Find help for a package.

More about an object

str(iris)
Get a summary of an object's structure.
class(iris)
Find the class an object belongs to.

Using Libraries

install.packages('dplyr')
Download and install a package from CRAN.

library(dplyr)
Load the package into the session, making all its functions available to use.

dplyr::select
Use a particular function from a package.

data(iris)
Load a built-in dataset into the environment.

Working Directory

getwd()
Find the current working directory (where inputs are found and outputs are sent).

setwd('C://file/path')
Change the current working directory.

Use projects in RStudio to set the working directory to the folder you are working in.

Vectors

Creating Vectors

c(2, 4, 6)	2 4 6	Join elements into a vector
2:6	2 3 4 5 6	An integer sequence
seq(2, 3, by=0.5)	2.0 2.5 3.0	A complex sequence
rep(1:2, times=3)	1 2 1 2 1 2	Repeat a vector
rep(1:2, each=3)	1 1 1 2 2 2	Repeat elements of a vector

Vector Functions

sort(x)
Return x sorted.
rev(x)
Return x reversed.
table(x)
See counts of values.
unique(x)
See unique values.

Selecting Vector Elements

By Position

x[4]	The fourth element.
x[-4]	All but the fourth.
x[2:4]	Elements two to four.
x[-(2:4)]	All elements except two to four.
x[c(1, 5)]	Elements one and five.

By Value

x[x == 10]	Elements which are equal to 10.
x[x < 0]	All elements less than zero.
x[x %in% c(1, 2, 5)]	Elements in the set 1, 2, 5.

Named Vectors

x['apple']	Element with name 'apple'.

Programming

For Loop

```
for (variable in sequence){
   Do something
}
```

Example

```
for (i in 1:4){
   j <- i + 10
   print(j)
}
```

While Loop

```
while (condition){
   Do something
}
```

Example

```
while (i < 5){
   print(i)
   i <- i + 1
}
```

If Statements

```
if (condition){
   Do something
} else {
   Do something different
}
```

Example

```
if (i > 3){
   print('Yes')
} else {
   print('No')
}
```

Functions

```
function_name <- function(var){
   Do something
   return(new_variable)
}
```

Example

```
square <- function(x){
   squared <- x*x
   return(squared)
}
```

Reading and Writing Data

Input	Ouput	Description
df <- read.table('file.txt')	write.table(df, 'file.txt')	Read and write a delimited text file.
df <- read.csv('file.csv')	write.csv(df, 'file.csv')	Read and write a comma separated value file. This is a special case of read.table/ write.table.
load('file.RData')	save(df, file = 'file.Rdata')	Read and write an R data file, a file type special for R.

Conditions

a == b	Are equal	a > b	Greater than	a >= b	Greater than or equal to	is.na(a)	Is missing
a != b	Not equal	a < b	Less than	a <= b	Less than or equal to	is.null(a)	Is null

https://iqss.github.io › dss-workshops › R › Rintro

Spatial manipulation with sf: : CHEAT SHEET

The sf package provides a set of tools for working with geospatial vectors, i.e. points, lines, polygons, etc.

Geometric confirmation

st_contains(x, y, ...) Identifies if x is within y (i.e. point within polygon)

st_covered_by(x, y, ...) Identifies if x is completely within y (i.e. polygon completely within polygon)

st_covers(x, y, ...) Identifies if any point from x is outside of y (i.e. polygon outside polygon)

st_crosses(x, y, ...) Identifies if any geometry of x have commonalities with y

st_disjoint(x, y, ...) Identifies when geometries from x do not share space with y

st_equals(x, y, ...) Identifies if x and y share the same geometry

st_intersects(x, y, ...) Identifies if x and y geometry share any space

st_overlaps(x, y, ...) Identifies if geometries of x and y share space, are of the same dimension, but are not completely contained by each other

st_touches(x, y, ...) Identifies if geometries of x and y share a common point but their interiors do not intersect

st_within(x, y, ...) Identifies if x is in a specified distance to y

Geometric operations

st_boundary(x) Creates a polygon that encompasses the full extent of the geometry

st_buffer(x, dist, nQuadSegs) Creates a polygon covering all points of the geometry within a given distance

st_centroid(x, ..., of_largest_polygon) Creates a point at the geometric centre of the geometry

st_convex_hull(x) Creates geometry that represents the minimum convex geometry of x

st_line_merge(x) Creates linestring geometry from sewing multi linestring geometry together

st_node(x) Creates nodes on overlapping geometry where nodes do not exist

st_point_on_surface(x) Creates a point that is guarenteed to fall on the surface of the geometry

st_polygonize(x) Creates polygon geometry from linestring geometry

st_segmentize(x, dfMaxLength, ...) Creates linesting geometry from x based on a specified length

st_simplify(x, preserveTopology, dTolerance) Creates a simplified version of the geometry based on a specified tolerance

Geometry creation

st_triangulate(x, dTolerance, bOnlyEdges) Creates polygon geometry as triangles from point geometry

st_voronoi(x, envelope, dTolerance, bOnlyEdges) Creates polygon geometry covering the envolope of x, with x at the centre of the geometry

st_point(x, c(numeric vector), dim = "XYZ") Creating point geometry from numeric values

st_multipoint(x = matrix(numeric values in rows), dim = "XYZ") Creating multi point geometry from numeric values

st_linestring(x = matrix(numeric values in rows), dim = "XYZ") Creating linestring geometry from numeric values

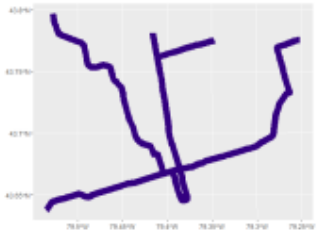
st_multilinestring(x = list(numeric matricesin rows), dim = "XYZ") Creating multi linestring geometry from numeric values

st_polygon(x = list(numeric matrices in rows), dim = "XYZ") Creating polygon geometry from numeric values

st_multipolygon(x = list(numeric matrices in rows), dim = "XYZ") Creating multi polygon geometry from numeric values

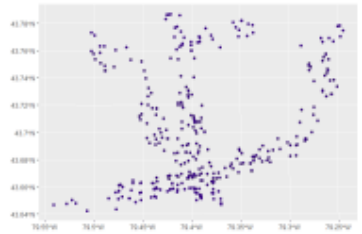
```
ggplot() +
  geom_sf(data = schools)
```

+



```
ggplot() +
  geom_sf(data = subway)
```

=>



```
ggplot() +
  geom_sf(data = st_intersection(schools, st_buffer(subway, 1000)))
```

This cheatsheet presents the sf package [Edzer Pebesma 2018] in version 0.6.3. See https://github.com/r-spatial/sf for more details.

Spatial manipulation with sf: : CHEAT SHEET

The sf package provides a set of tools for working with geospatial vectors, i.e. points, lines, polygons, etc.

Geometry operations

st_contains(x, y, ...) Identifies if x is within y (i.e. point within polygon)

st_crop(x, y, ..., xmin, ymin, xmax, ymax) Creates geometry of x that intersects a specified rectangle

st_difference(x, y) Creates geometry from x that does not intersect with y

st_intersection(x, y) Creates geometry of the shared portion of x and y

st_sym_difference(x, y) Creates geometry representing portions of x and y that do not intersect

st_snap(x, y, tolerance) Snap nodes from geometry x to geometry y

st_union(x, y, ..., by_feature) Creates multiple geometries into a a single geometry, consisiting of all geometry elements

Geometric measurement

st_area(x) Calculate the surface area of a polygon geometry based on the current coordinate reference system

st_distance(x, y, ..., dist_fun, by_element, which) Calculates the 2D distance between x and y based on the current coordinate system

st_length(x) Calculates the 2D length of a geometry based on the current coordinate system

Misc operations

st_as_sf(x, ...) Create a sf object from a non-geospatial tabular data frame

st_cast(x, to, ...) Change x geometry to a different geometry type

st_coordinates(x, ...) Creates a matrix of coordinate values from x

st_crs(x, ...) Identifies the coordinate reference system of x

st_join(x, y, join, FUN, suffix, ...) Performs a spatial left or inner join between x and y

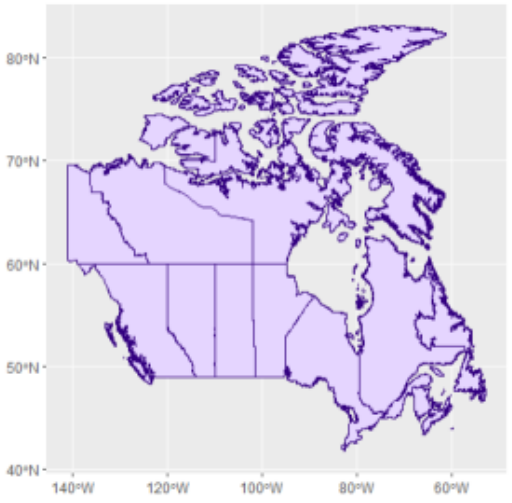
st_make_grid(x, cellsize, offset, n, crs, what) Creates rectangular grid geometry over the bounding box of x

st_nearest_feature(x, y) Creates an index of the closest feature between x and y

st_nearest_points(x, y, ...) Returns the closest point between x and y

st_read(dsn, layer, ...) Read file or database vector dataset as a sf object

st_transform(x, crs, ...) Convert coordinates of x to a different coordinate reference system



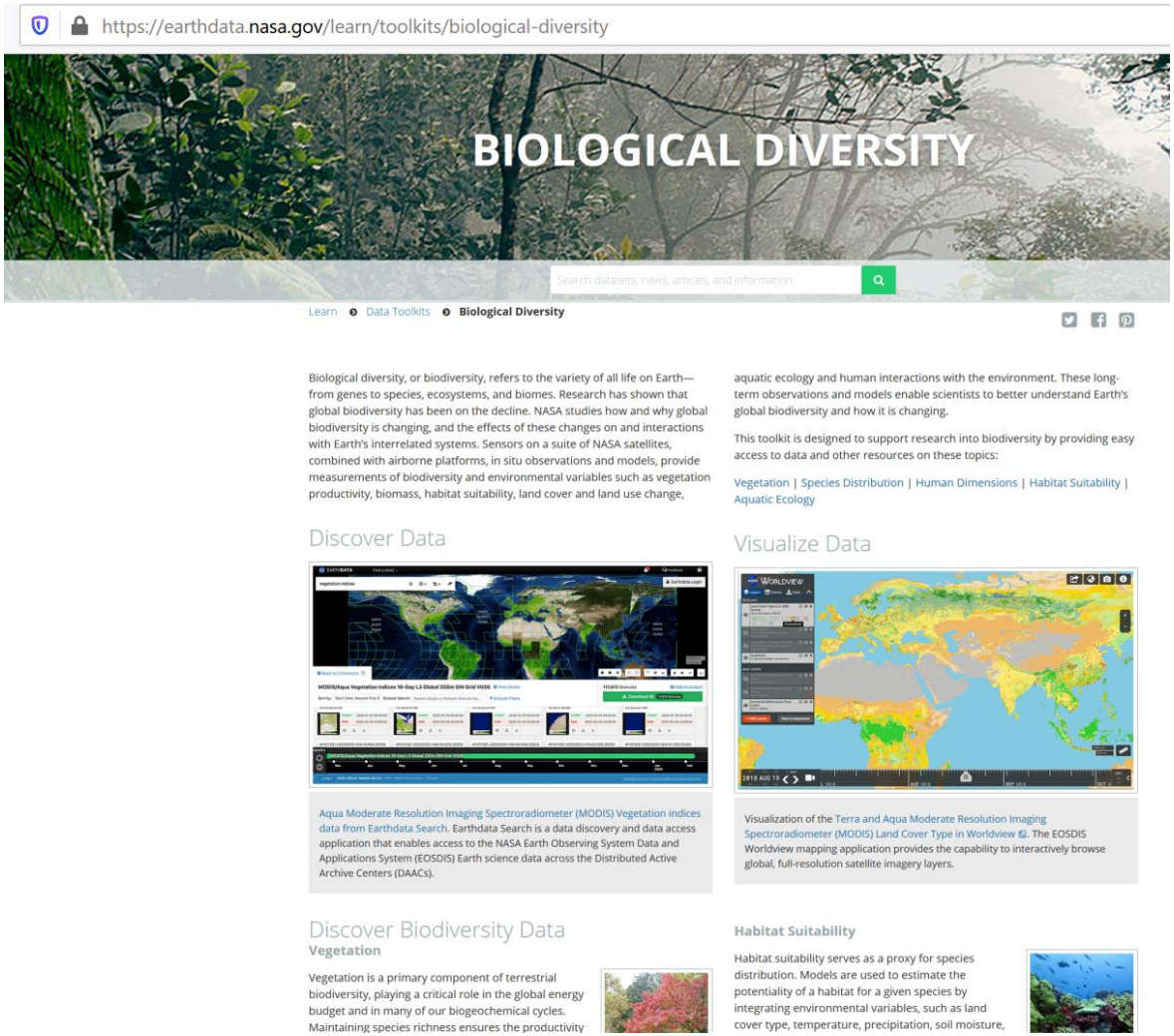

```
ggplot() +
  geom_sf(data = cdn) +
  coord_sf(crs = st_crs(4326))
```


```
ggplot() +
  geom_sf(data = cdn) +
  coord_sf(crs = st_crs(3347))
```

Coordinates projection on the fly

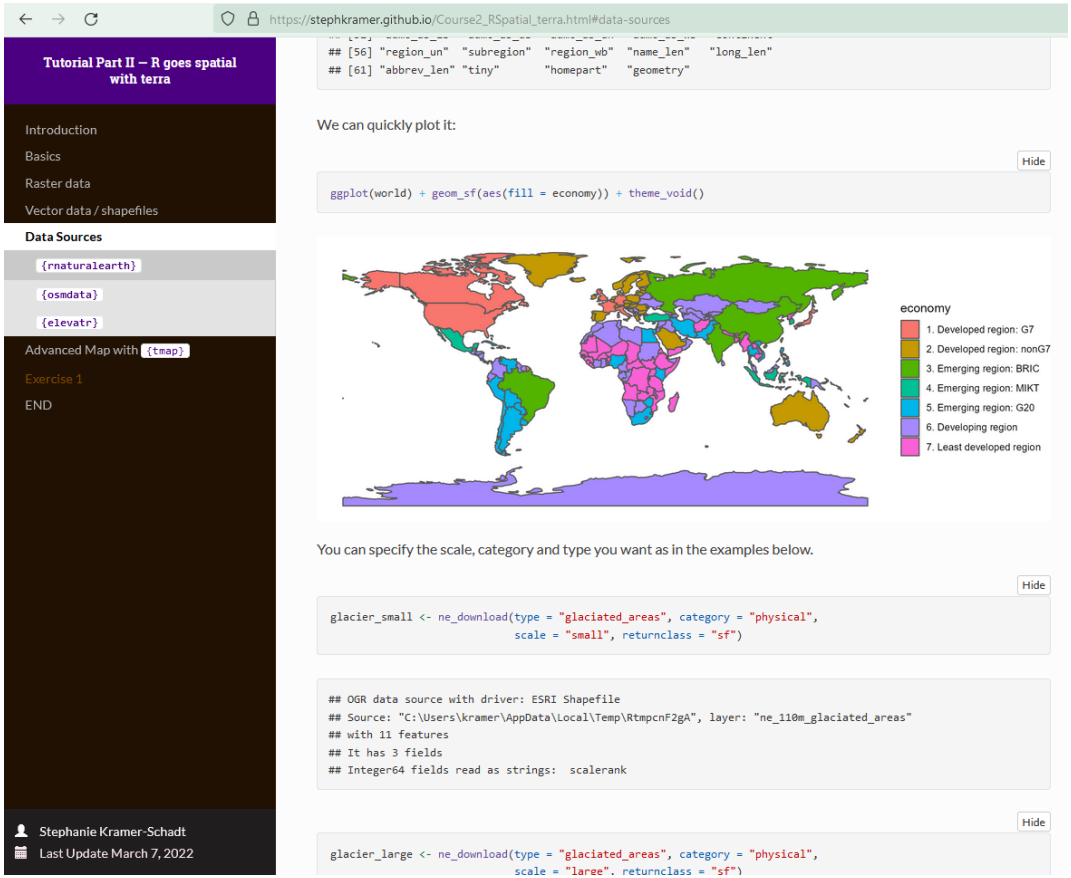
Free geodata

- https://earthdata.nasa.gov/learn/toolkits/biological-diversity



Free data in R

- rnaturalearth
- terrainr (and elevatr, rayvista,…)
- osmdata



Cedric has added them to the course 2 file!

Leibniz Institute for Zoo and Wildlife Research
IN THE FORSCHUNGSVERBUND BERLIN E.V.

Member of the Leibniz Association

Let‘s start

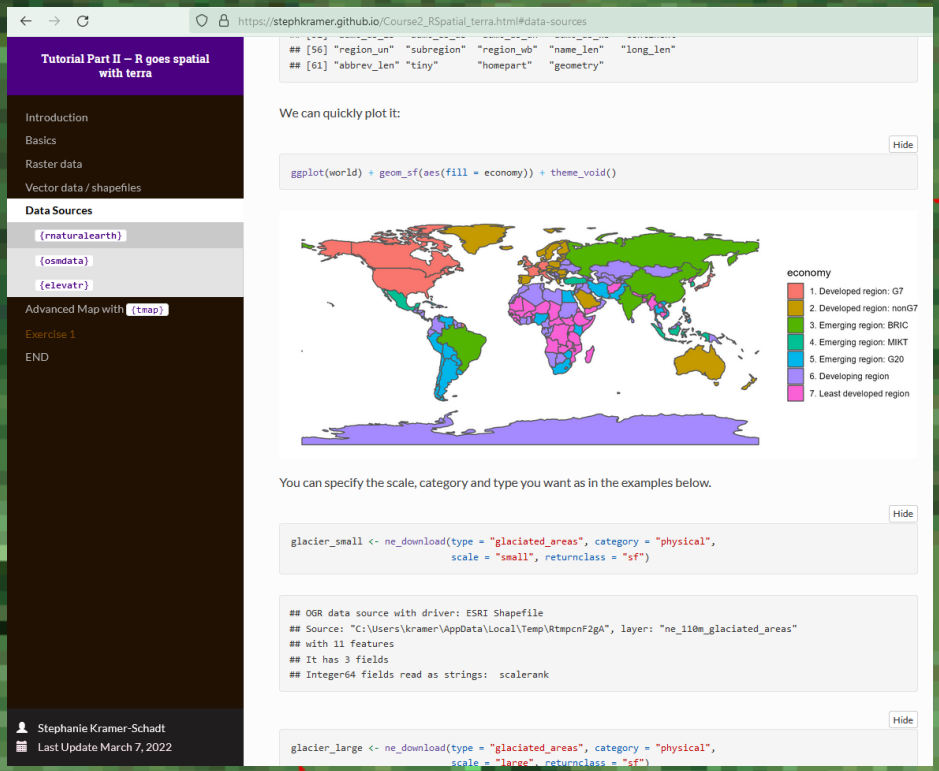
https://www.berliner-kurier.de/berlin/kiez--stadt/achterbahn-wetter-sommer-startet-mit-temperatursturz-30637472

Exercises 1 (pre-course preparation)

Leibniz Institute for Zoo and Wildlife Research
IN THE FORSCHUNGSVERBUND BERLIN E.V.
Member of the Leibniz Association

RMarkdown

Rstudio -> File -> New File –> R Markdown… -> ok



https://bookdown.org/yihui/rmarkdown/

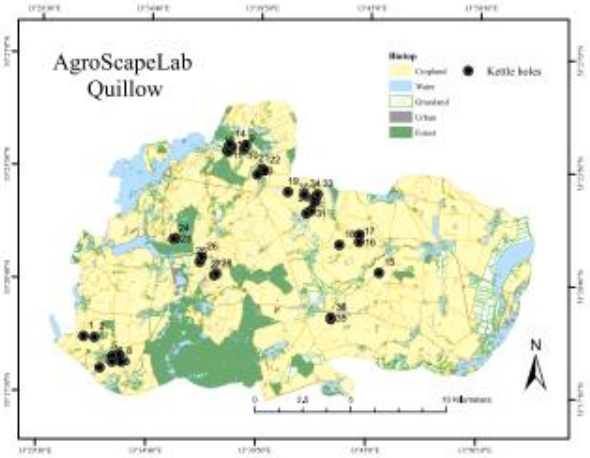
Contents lists available at ScienceDirect

Agriculture, Ecosystems and Environment

journal homepage: www.elsevier.com/locate/agee

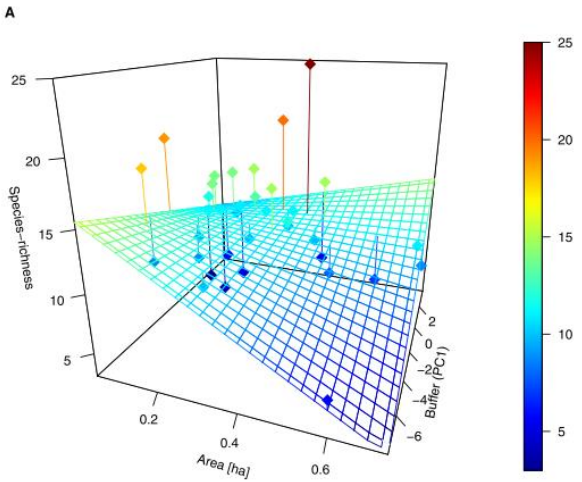
Habitat quality and connectivity in kettle holes enhance bee diversity in agricultural landscapes

Sissi Lozada-Gobilard [a,b,*,1,2], Carlos Miguel Landivar Albis [c], Karolina Beata Rupik [d], Marlene Pätzig [e,3], Sebastian Hausmann [f,g], Ralph Tiedemann [b], Jasmin Joshi [g,h,4]



- Kettle hole density in different buffers
- Number of habitat types around kettle holes
- Tree cover density

S. Lozada-Gobilard et al.



Course 2 Exercise 2

- Little helpers for projects in Berlin:
- d6berlin

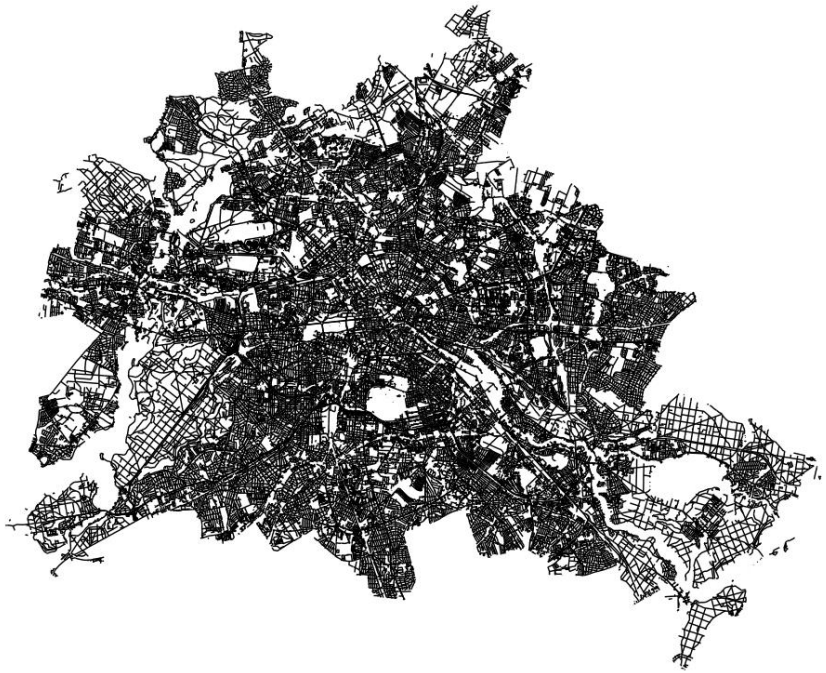
Leibniz Institute for Zoo and Wildlife Research
IN THE FORSCHUNGSVERBUND BERLIN E.V.

Member of the Leibniz Association

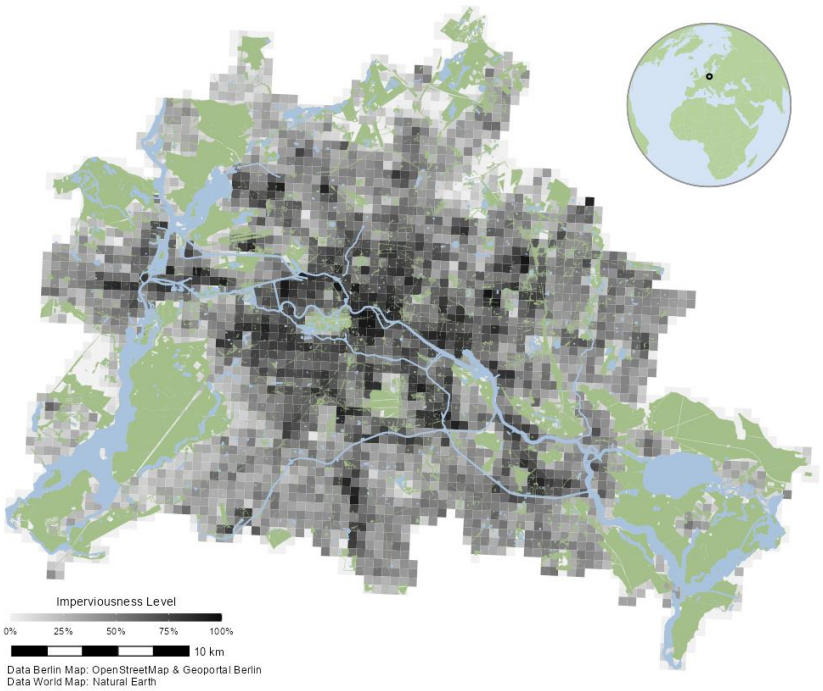
The {d6berlin} R Package

The {d6berlin} package aims to provide **spatial data and template maps for Berlin**

`d6berlin::sf_roads`



`d6berlin::base_map_imp()`



The {d6berlin} R Package

The {d6berlin} package aims to provide **spatial data and template maps for Berlin**.

```
install.packages("devtools")
devtools::install_github("EcoDynIZW/d6berlin")
```

```
library(d6berlin)
names(sf_green)
[1] "osm_id" "code" "fclass" "name" "district_name" "district_id" "district_key" "geometry“

unique(sf_green$fclass)
[1] "scrub" "grass" "forest" "meadow" "nature_reserve" "beach" "heath" "cliff"

sf_forest <- subset(sf_green, fclass == "forest")
```

The {d6berlin} R Package

The {d6berlin} package aims to provide **spatial data and template maps for Berlin**.

```
install.packages("devtools")
devtools::install_github("EcoDynIZW/d6berlin")
```

```
library(d6berlin)
library(ggplot2)

ggplot(sf_forest) + geom_sf()
```

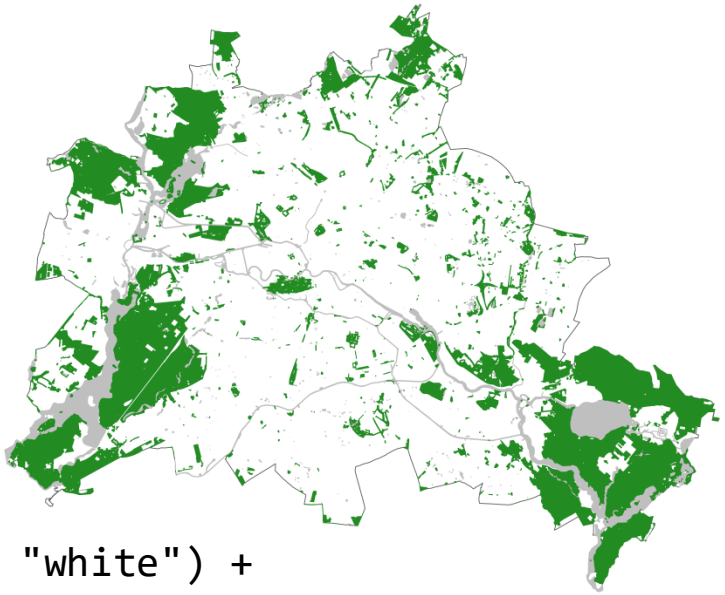
The {d6berlin} R Package

The {d6berlin} package aims to provide **spatial data and template maps for Berlin**.

```
install.packages("devtools")
devtools::install_github("EcoDynIZW/d6berlin")
```

```
library(d6berlin)
library(ggplot2)

ggplot(sf_forest) + geom_sf()
```



```
ggplot() +
  geom_sf(data = sf_berlin, fill = "white") +
  geom_sf(data = sf_water,  fill = "grey75", color = "grey75") +
  geom_sf(data = sf_forest, fill = "forestgreen", color = "forestgreen")
```

Exercises

- https://github.com/stephkramer/Course2_RSpatial

README.md

Using our long standing projects in Borneo, we demonstrate how to use R as a GIS. A precondition for following this course is knowing well the basics of R, as summarised in R Intro. Please download the course data here (approx. 600 MB):

https://www.dropbox.com/sh/91qey7u5ohn5fmf/AAD84ZxnExw4Chd93XidEaF6a?dl=0

and then follow the course here: R goes Spatial

As a 'learning control exercise', please solve Exercise 1 in course 2 material yourself (C2_E1).

More exercises here:

- Course 2 Exercise 2 C2_E2
- Course 2 Exercise 3 C2_E3

Let‘s start in Breakout Rooms

Leibniz Institute for Zoo and Wildlife Research
IN THE FORSCHUNGSVERBUND BERLIN E.V.
Member of the Leibniz Association

Course 2 Exercise 2

Exercise 2.1

Using data on green spaces in Berlin, estimate the number of wild boars observed inside green areas that are 1 ha or larger. What is the proportion of wild boars outside large green areas? Visualize the observations on a map showing the green spaces and the observations encoded by their location inside or outside the green spaces.

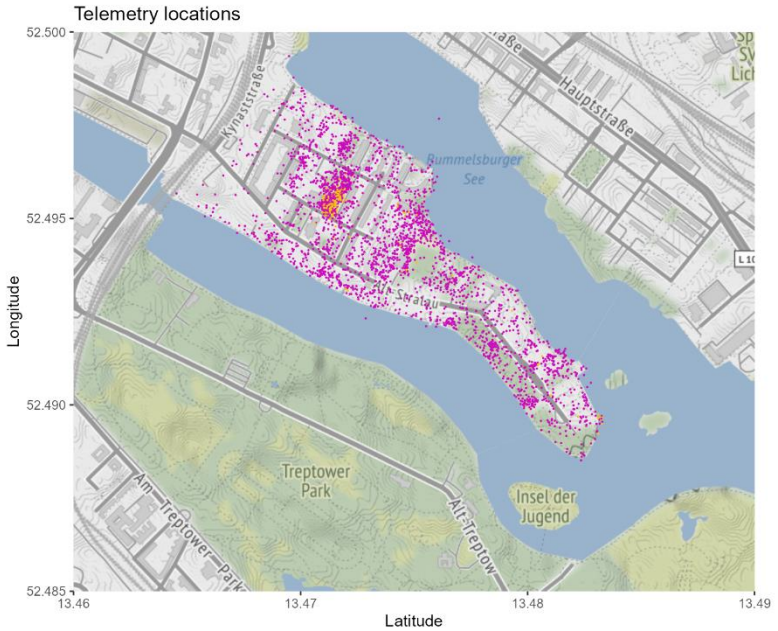
Individual Steps:

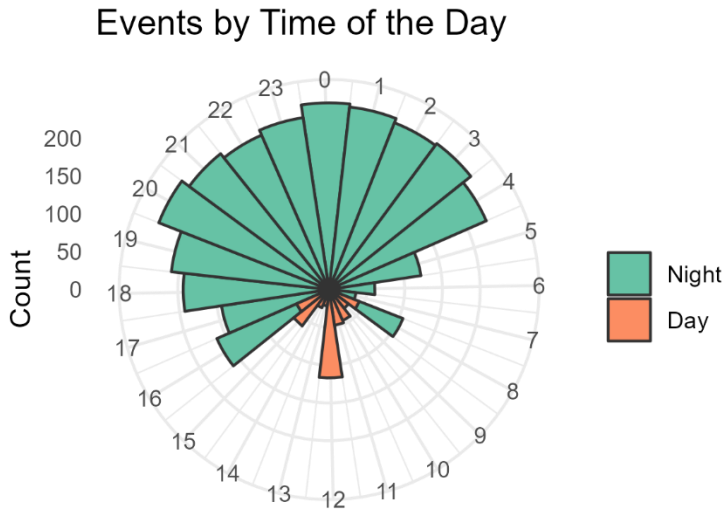
- 1. Store the simple feature data `sf_green` which is provided by the `{d6berlin}` package in a local object.
- 2. Calculate the area for each green space polygon and filter for green spaces that are equal or greater than 1 ha.
- 3. Load the wild boar data (`data_wb_melden_en.csv`) and turn it into a simple feature object. Remember to set the correct CRS!
- 4. Intersect the wild boar data with the filtered green spaces and extract the number of unique observations.
- 5. Use the new data to visualize observations inside and outside the green spaces that are 1 ha or larger as a map.


Leibniz Institute for Zoo
and Wildlife Research
IN THE FORSCHUNGSVERBUND BERLIN E.V.


Member of the
Leibniz
Leibniz Association

Sneek preview: Day 2 Introduction to movement analysis

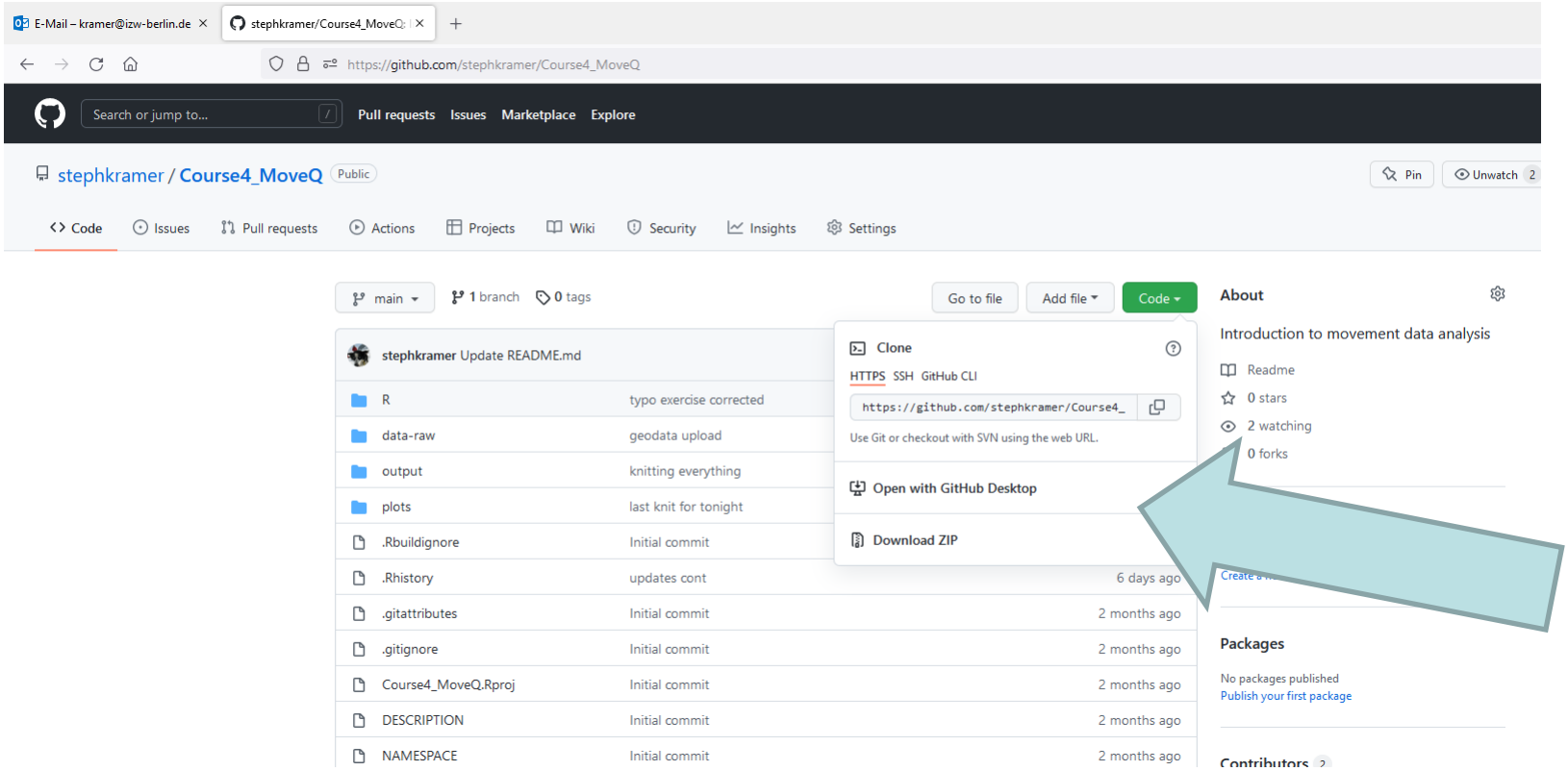




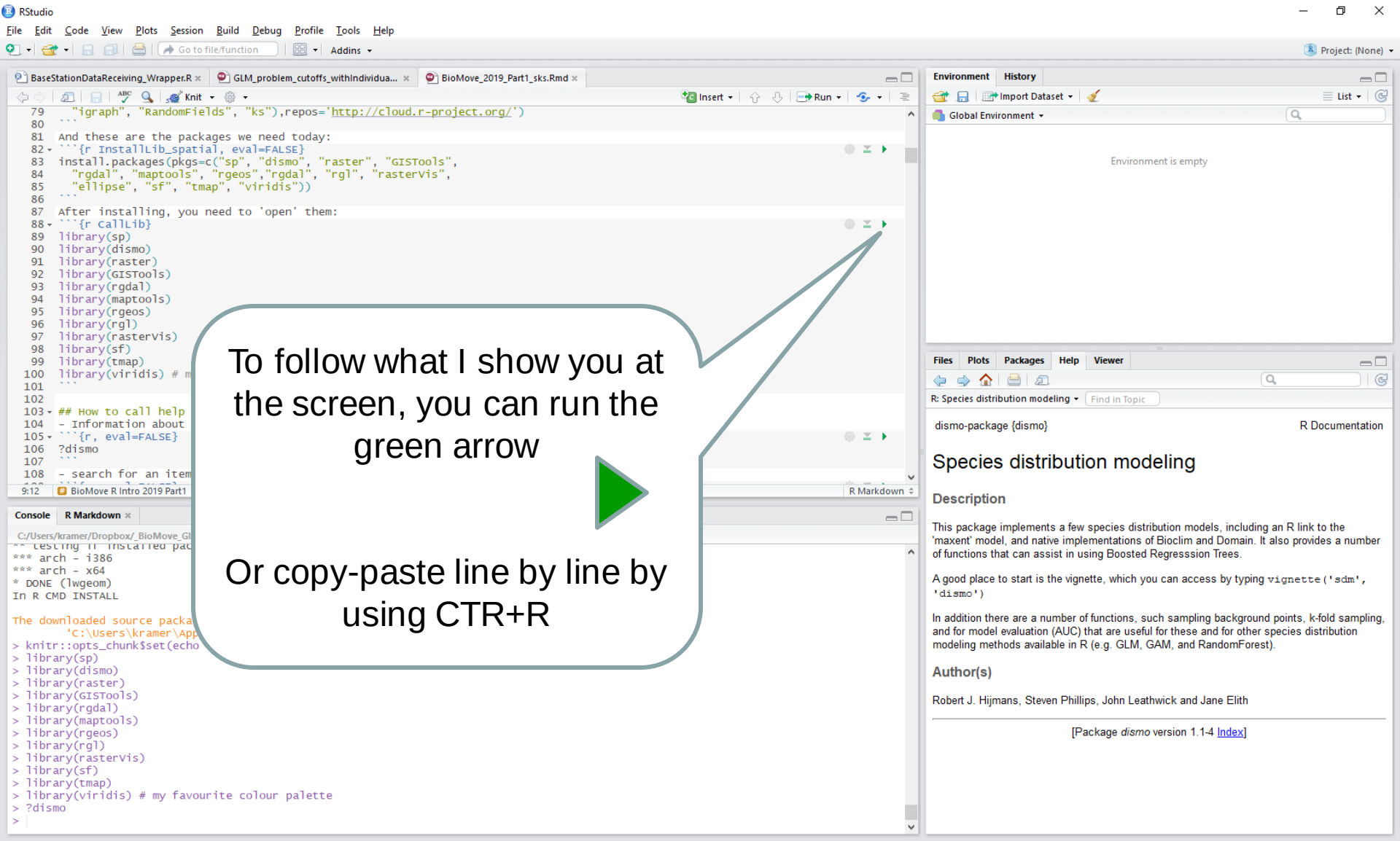
Day 2: Intro to Movement Data

https://github.com/stephkramer/Course4_MoveQ

Clone or download and unzip the repository



R-Studio/ File/ Rmarkdown -> rmd-file



To follow what I show you at the screen, you can run the green arrow

Or copy-paste line by line by using CTR+R


END of day 1