

Stichpunkte zur technischen Ausarbeitung der interaktiven Karte (interactiveMap.R):

Verwendete libraries:

- Wichtigste libraries: leaflet (für die interaktive karte selbst) und shiny (für das hosting); sind denke ich nich weiter erklärungsbedürftig, da für diese Funktionen wohl der Standard
- sampSurf ist für die Wohnumkreis-NDVI Funktion (erlaubt es ein spatial Polygon zu erstellen mit eingestelltem radius um den angeklickten punkt)

Vorgehen:

1) data frames aufbauen:

- zunächst werden die shapefiles von den Bezirken Leipzigs über die Funktion readOGR() eingelesen (wo die shapes genau her sind müsstet ihr nochmal Tom fragen, glaube direkt von dem Leipziger Datenportal auch)
- die shapefiles enthalten die Geodaten der Bezirke, sodass sie in R nach dem einlesen nun als geospatial objects vorliegen (in variable leipzig_Bezirke)
- dann wird auf diese Objekte die Funktion spTransform angewandt, was die Projektion von Koordinaten in Winkel (Longitude und Latitude) ändert; die shapes sind also jetzt über lng und lat beschrieben (leaflet erwartet das so, daher ist der schritt nötig)
- dann werden die vorher von tom mittels VegetationIndex.R berechneten NDVI und EVI Werte aus den entsprechenden .csv Dateien eingelesen und zum data frame hinzugefügt; in der Variable Bezirksdaten_insgesamt liegen dann also alle nötigen Daten (Geodaten der Bezirke und jeweils NDVI, EVI dazu) vor
- analog wird dann der data frame für die Ortsteile aufgebaut, bis Ortsteildaten_gesamt vorliegt

2) daten mittels leaflet als interaktive karte darstellen und mittels shiny hosten:

- das meiste hier ist einfach formatierung und nicht wirklich interessant
- in ‚ui‘ sind die user funktionalitäten und das grundsätzliche layout hinterlegt (action buttons und checkBoxes um zwischen Ortsteilen und Bezirken zu wechseln, die Legende ein- und auszublenden, in den Modus für den Wohnumkreis-NDVI zu schalten und den Radius für diesen Wohnumkreis einstellen zu können); würde ich alles höchstens knapp erwähnen, nur die Wohnumkreis-Funktion ist erklärungsbedürftig
- in ‚server‘ wird dann die eigentliche output map mittels der funktion renderLeaflet() erstellt; auch hier definiert der code viel das layout; außerdem gibt es diverse observer; das sind funktionen aus shiny, welche auf user input warten und entsprechend die karte ändern (z.Bsp. ob Ortsteile oder Bezirke angezeigt werden)
- in einem dieser observer ist unsere Wohnumkreis-Funktion definiert, das funktioniert so:

wenn wir uns im entsprechenden modus befinden (der user muss hierfür zunächst einen haken in einer checkbox setzen), wird auf einen klick des users auf die map gewartet; von diesem klick werden longitude und latitude genommen und wiederum in eine andere projektion transformiert (das ist nötig, weil die .tif-Dateien, also die Raster, auf denen die NDVI bzw. EVI Berechnungen basieren, das so brauchen; für genauere Details fragt bitte nochmal Tom, das habe ich auch nur so halb verstanden an der stelle); dann wird mittels der funktion `spCircle` aus der library `sampSurf` ein kreisförmiges `spatialPolygon` ((siehe bei bedarf die dokumentation hier:

<https://www.rdocumentation.org/packages/sampSurf/versions/0.7-6/topics/spCircle>)

um diese Koordinaten herum erstellt (den radius kann der user mittels slider vorher einstellen; default ist 1000m)

dann wird dieses kreisförmige `spatial polygon` in die funktion `compute_vegetation_index()` von tom gegeben, um für den kreis den ndvi zu berechnen; dann wird mittels der funktion `addCircles()` ein Kreis auf die Karte gemalt, für den der entsprechende NDVI als Grünton auf unserer Farbskala die Füllfarbe ist; außerdem wird wie für die shapes auch der NDVI beim anklicken und drüberfahren mit der Maus als Wert angezeigt

-
- das wäre es im wesentlichen, ich hoffe das war halbwegs verständlich; ansonsten hilft auch ein blick in das R-Skript, da ist auch alles entsprechend kommentiert
 - als Quelle um was für Leaflet oder Shiny nachzuschlagen kann ich die Seite hier empfehlen:

<https://rstudio.github.io/leaflet/projections.html>