Defining Geographic Regions of Interest with Geocomputation and R

Marcelo de Carvalho Alves

07 agosto, 2025

Homework 1

Create a polygon characterizing an area of interest

The objective of the homework assignment is to create a polygon characterizing an area of interest that can be used in the practical work of the subject. The polygon should be created in Google Earth by drawing it on top of a remote sensing image with very high spatial resolution. This enables observation of details around the area where the polygon was created. After creating the polygon in Google Earth, configuring a closed geometric figure, it must be imported into R and mapped to make a first map in the course. Suggested systematic steps to perform the task are given below.

Defining geographical region from polygon drawn in Google Earth

A polygon created in Google Earth with the `KML` extension can be imported into R as a vector that refers to an object or remote sensing process defined in relation to some location on Earth by creating a polygon drawn on top of the target of interest, as in a color composite satellite image.

A version of Google Earth Pro for Windows can be obtained from **download**.

To create a polygon in Google Earth, (1) open Google Earth. (2) Go to a location on the map. (3) Above the map, click Add Path. (4) To add a shape, click `Add Polygon`. (5) A `New Polygon` dialog box will appear. (6) To draw the shape you want using connected lines, click on a starting point on the map and drag it. (7) Click on an end point to close the polygon. Once you have a closed polygon geometry, you can calculate the area of the polygon by going to `Tools` <- `Ruler`. However, calculating the area is not our objective at the moment. (8) Enter a description and properties for the polygon. The polygon will be created as a layer in the left menu of Google Earth. (9) Click on the polygon layer created with the right mouse button and export the file with the extension `nameoffile.kml` with the option `save place as`. (10) Click on `save`.

Mapping KML polygons in R

Packages used

The R software is used as an example, but other software can be used to perform this task.

The `library` function in the R console is used to load the packages needed for this data analysis.

```
library(sf)
library(dplyr)
library(mapview)
```

Import KML file into R

Polygon geometries related to the boundaries created on very high spatial resolution images from Google Earth are used. The files are imported into R using the `st_read` function.

```
L29 <- st_read("C:/TCC/VitorFerreira/Lavoura_29.kml")
head(L29)
```

```
#Simple feature collection with 1 feature and 2 fields
#Geometry type: POLYGON
#Dimension:     XYZ
#Bounding box:  xmin: -44.96336 ymin: -21.22896 xmax: -44.96173 ymax: -21.22843
#z_range:       zmin: 0 zmax: 0
#Geodetic CRS:  WGS 84
#        Name Description                       geometry
#1 Lavoura 29             POLYGON Z ((-44.96173 -21.2...
L36 <- st_read("C:/TCC/VitorFerreira/Lavoura_36.kml")
head(L36)
#Simple feature collection with 1 feature and 2 fields
#Geometry type: POLYGON
#Dimension:     XYZ
#Bounding box:  xmin: -44.96357 ymin: -21.23146 xmax: -44.96134 ymax: -21.22897
#z_range:       zmin: 0 zmax: 0
#Geodetic CRS:  WGS 84
#        Name Description                       geometry
#1 Lavoura 36             POLYGON Z ((-44.96134 -21.2...
```

Remove the Z dimension

The Z geometry obtained in the polygon is not needed. The `st_zm` function is used with the `drop=TRUE` argument in order to remove the Z dimension from the file, which is necessary for subsequent mapping.

```
L29zm<-st_zm(L29, drop = TRUE)
L29zm
#Simple feature collection with 1 feature and 2 fields
#Geometry type: POLYGON
#Dimension:     XY
#Bounding box:  xmin: -44.96336 ymin: -21.22896 xmax: -44.96173 ymax: -21.22843
#Geodetic CRS:  WGS 84
#        Name Description                       geometry
#1 Lavoura 29             POLYGON ((-44.96173 -21.228...
L36zm<-st_zm(L36, drop = TRUE)
L36zm
#Simple feature collection with 1 feature and 2 fields
#Geometry type: POLYGON
#Dimension:     XY
#Bounding box:  xmin: -44.96357 ymin: -21.23146 xmax: -44.96134 ymax: -21.22897
#Geodetic CRS:  WGS 84
#        Name Description                       geometry
#1 Lavoura 36             POLYGON ((-44.96134 -21.229...
```

Mapping the polygons

The `mapview` function is used to map coffee crops on the campus of the Federal University of Lavras, Brazil. A template from the OpenStreetMap database is used as a geovisualization option added to the performed mapping (Figure @ref(fig:fig24a)).

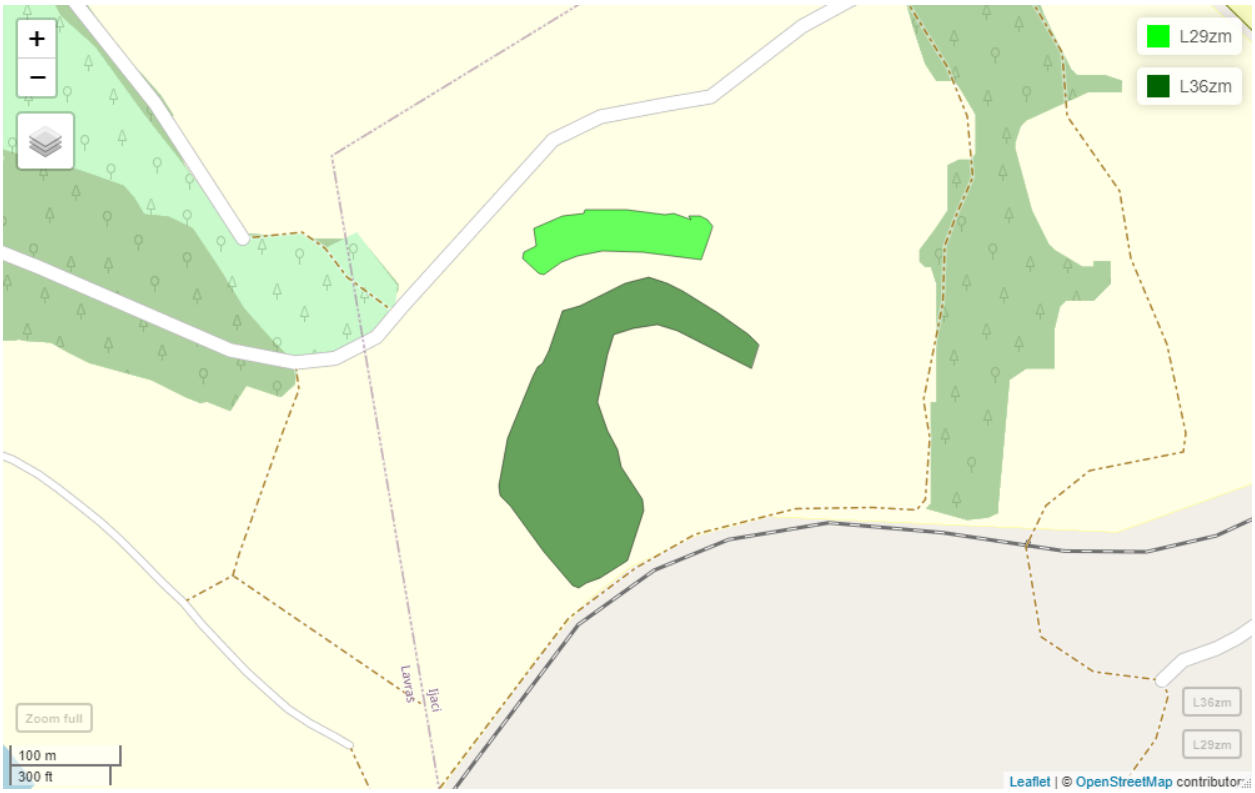

Figure 1: Mapping coffee crops mapped using geocomputation techniques in Google Earth and R.

```
mapview(list(L29zm,L36zm), col.regions=list("green", "DarkGreen"),
        col=list("black","black"))
```

Convert polygons into a single file

The polygons are converted into a single file of simple polygon features with attributes. The `bind_rows` function of the `dplyr` package is used to list all the polygons in a single `sf` file.

```
single_sf <- dplyr::bind_rows(list(L29zm,L36zm))
```

The columns `id` and `name` are created and the bracket operator is used to select only the columns created and the geometry in a single file called `polsf`. This polygon is used as base for subsequent mappings.

```
single_sf$id = c(1:2)
single_sf$name = c("Lavoura 36","Lavoura 29")
single_sf
#Simple feature collection with 2 features and 4 fields
#Geometry type: POLYGON
#Dimension:     XY
#Bounding box:  xmin: -44.96357 ymin: -21.23146 xmax: -44.96134 ymax: -21.22843
#Geodetic CRS:  WGS 84
#        Name Description                       geometry id       name
#1 Lavoura 29             POLYGON ((-44.96173 -21.228...  1 Lavoura 36
#2 Lavoura 36             POLYGON ((-44.96134 -21.229...  2 Lavoura 29
```

Export the polygon as ESRI shapefile

Discrete vector objects with a geographic database are exported to the directory of interest with the `st_write` function and the file name `pols`.

```
st_write(single_sf, dsn = "C:/Aulas/2021_2/GEO/shp/pols.shp", layer = "pols.shp",
         driver = "ESRI Shapefile")
#Warning: Field names abbreviated for ESRI Shapefile driverWriting layer `pols'
#to data source #`C:/Aulas/2021_2/GEO/shp/pols.shp' using driver `ESRI Shapefile'
#Warning: GDAL Message 6: Normalized/laundered field name: 'name' to 'name_1'
#Writing 2 features with 4 fields and geometry type Polygon.
```

Other options can be used to create and manipulate vectors in R.

Manipulate vector in R's terra package

Functions from the `terra` package are used below to import and map the vector file created in Google Earth.

Loading terra package

Since the `terra` package is already installed in the computer, the `library` function is used to load the package.

```
library(terra)
#terra 1.7.3
```

Importing polygons into R

The `vect` function is used to import the shapefile exported to the directory of interest with `sf` package functions.

```
pols <- vect("C:/Aulas/2021_2/GEO/shp/pols.shp")
```

Observe the file header

Note that the file class is `SpatVector`. The coordinate system is WGS 84.

```
pols
#class       : SpatVector
# geometry    : polygons
# dimensions  : 2, 4  (geometries, attributes)
# extent  : -44.96357, -44.96134, -21.23146, -21.22843 (xmin, xmax, ymin, ymax)
# source      : pols.shp
# coord. ref. : lon/lat WGS 84 (EPSG:4326)
# names       :       Name Dscrptn    id     name_1
# type        :      <chr>   <chr> <int>      <chr>
# values      : Lavoura 29      NA     1 Lavoura 36
#               Lavoura 36      NA     2 Lavoura 29
```

Map simple polygon feature in R

The polygon is mapped with the `plot` function (Figure @ref(fig:fig25a)).

```
plot(pols, col = "grey92")
```

Convert object of class SpatVector into sf and vice versa

Objects of class `SpatVector` can be converted into class `sf` and vice versa with the functions `st_as_sf` and `vect`, respectively.

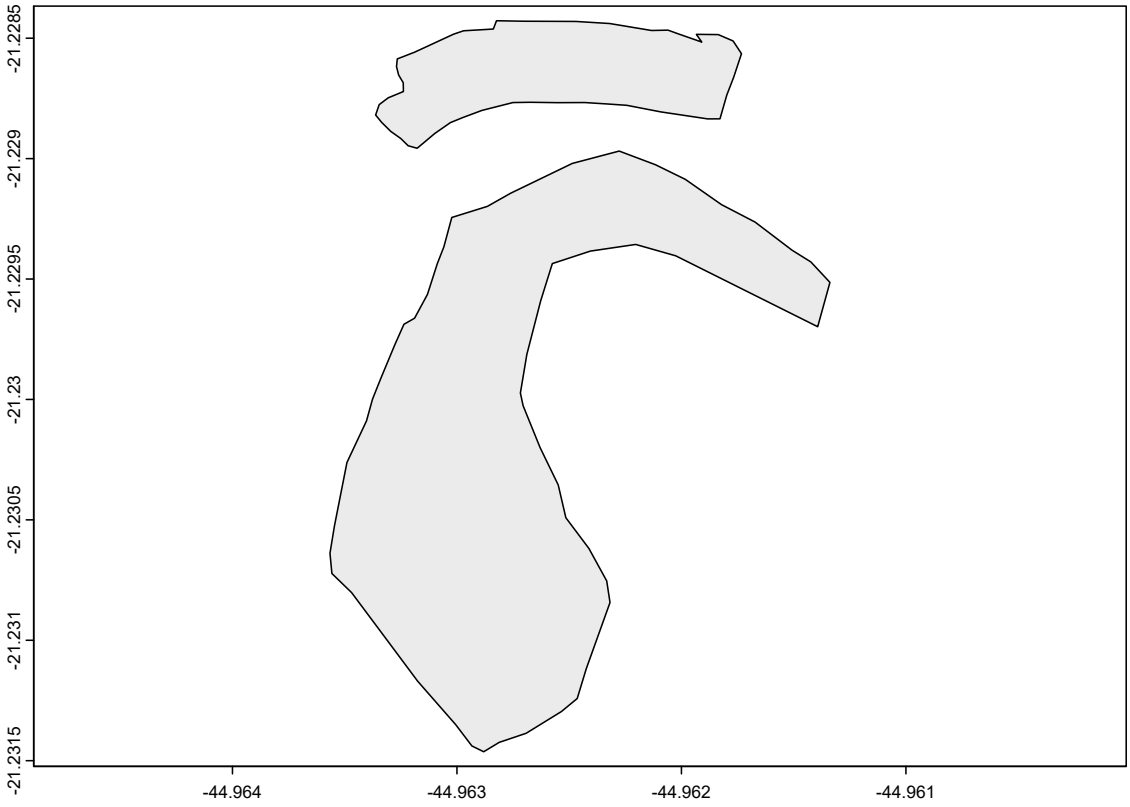

Figure 2: Mapping a geographical region using the terra package.

```r
polsf <- st_as_sf(pols) # Convert to sf data frame class
class(polsf)
#[1] "sf"         "data.frame"
polvect <- vect(polsf) # Convert to SpatVector class
class(polvect)
#[1] "SpatVector"
#attr(,"package")
#[1] "terra"
```