extRatum package example

Nikos Patias

12/12/2020

This notebook provides an example of `extRatum` package

This is a notebook that demonstrates the use of `extRatum` package, using OpenStreetMap data. `extRatum` package can be used to summarise the presence of geospatial features in larger geographic areas. It does so by calculating the area covered by a geospatial feature (i.e. buildings, parks, lakes, etc.). The geospatial features can be of all types of geospatial data (i.e. point, polygons or lines).

In this example, we focus on built environment characteristics. We make use of OpenStreetMap data and calculate indicators for point, polygon and line data. The reference layer is the Lower Layer Super Output Area (LSOA) boundaries for the city of Liverpool in the United Kingdom.

```
# these are the packages needed
#devtools::install("C:/Users/User/Desktop/extRatum")
library(extRatum)
library(sf)
```

```
## Linking to GEOS 3.8.0, GDAL 3.0.4, PROJ 6.3.1
```

```
library(dplyr)
```

```
##
## Attaching package: 'dplyr'

## The following objects are masked from 'package:stats':
##
##     filter, lag

## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```
library(tmap)
library(osmdata)
```

```
## Data (c) OpenStreetMap contributors, ODbL 1.0. https://www.openstreetmap.org/copyright
```

Read Liverpool LSOA boundaries

First, we read in the LSOA boundaries for Liverpool. The data downloaded from CDRC website: https://data.cdrc.ac.uk/

```
# 1. Read in the FUA grids
LSOAs <- st_read("layers/E08000012.shp")
```

```
## Reading layer `E08000012' from data source `C:\Users\User\Desktop\built_env_r_package\layers\E080000
## Simple feature collection with 298 features and 1 field
## geometry type:  MULTIPOLYGON
## dimension:      XY
## bbox:           xmin: 333086.1 ymin: 381426.3 xmax: 345636 ymax: 397980.1
## projected CRS:  Transverse_Mercator
```

Because the area of interest is Liverpool, we should select the British National Grid as a planar coordinate system of reference throughout this notebook.

```
BNG = "epsg:27700"
```

Point data analysis

In this part of the notebook, we will deal with point data.

We create a simple query to download point data representing shops in Liverpool.

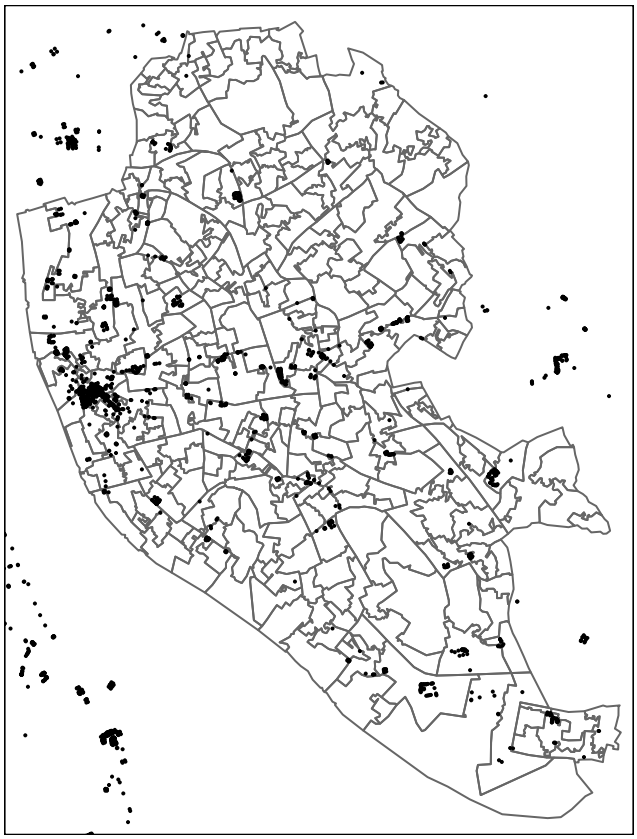
```
q <- getbb("Liverpool") %>%
  opq() %>%
  add_osm_feature(key = "shop")

shops <- osmdata_sf(q)
```

And plot them.

```
#tmap_mode("view") #use this code for creating an interactive map
tmap_mode("plot")
```

```
## tmap mode set to plotting
```

```
# show the points and grids
tm_shape(LSOAs) +
  tm_borders() +
  tm_shape(shops$osm_points) +
  tm_dots()
```



Then we calculate the number of points in each polygon using `point_calc()` function. Note that we have to pass a planar coordinate system in all our functions. Here we use British National Grid. The output of this function will be a dataframe containing:

- the LSOA code;
- the total area in sqkm of each LSOA;
- the number of points (i.e. shops) in each LSOA; and
- the ratio of points to the total LSOA area (or in other words the number of points by sqkm). In this way, we have a relative measure that can be compared across all LSOAs and is independent of their size.

Note that we have used the argument `total_points = TRUE` which returns the total number of points without differentiating between different shop types.

```
Shops_total <- point_calc(
  point_data = shops$osm_points,
  higher_geo_lay = LSOAs,
  unique_id_code = 'lsoa11cd',
  crs = BNG,
  total_points = TRUE
  )

# inspect the results
head(Shops_total)
```

```
## # A tibble: 6 x 4
```

```
##   lsoa11cd  Tot_area_sqkm NoPoints Ratio
##   <chr>             <dbl>    <int> <dbl>
## 1 E01006513         0.555       30 54.1
## 2 E01006514         0.262        2  7.63
## 3 E01006515         0.366       16 43.7
## 4 E01006518         0.235        1  4.25
## 5 E01006520         0.259        1  3.85
## 6 E01006522         0.474        3  6.33
```

In some cases though we want to know the split between different types of points. In this case, we change the `total_points = FALSE` and specify the column name that includes the classification (see `class_col`).

```
Shops_class <- point_calc(
  point_data = shops$osm_points,
  higher_geo_lay = LSOAs,
  unique_id_code = 'lsoa11cd',
  class_col = 'shop',
  crs = BNG,
  total_points = FALSE
  )
```

The output of this function will be a list of three dataframes.

1. A dataframe in long format reporting:

- the LSOA code;
- the total area in sqkm of each LSOA;
- the classification of the points within each LSOA;
- the number of points in each class (i.e. bakery, beauty) in each LSOA; and
- the ratio of points in each class to the total LSOA area (or in other words the number of points by sqkm).

```
head(Shops_class$PointsLong)
```

```
## # A tibble: 6 x 5
##   lsoa11cd  Tot_area_sqkm shop        NoPoints Ratio
##   <chr>             <dbl> <chr>          <int> <dbl>
## 1 E01006513         0.555 bakery             3  5.41
## 2 E01006513         0.555 beauty             1  1.80
## 3 E01006513         0.555 bookmaker          2  3.60
## 4 E01006513         0.555 books              2  3.60
## 5 E01006513         0.555 clothes            1  1.80
## 6 E01006513         0.555 convenience        7 12.6
```

2. A dataframe in wide format reporting:

- the LSOA code;
- the number of points in each class (i.e. bakery, beauty) in each LSOA.

```
head(Shops_class$PointsCountWide)
```

```
## # A tibble: 6 x 94
##   lsoa11cd alcohol  baby  Baby baby_goods   bag bakery beauty   bed betting
##   <chr>      <int> <int> <int>      <int> <int>  <int>  <int> <int>   <int>
## 1 E010065~      NA    NA    NA         NA    NA      3      1    NA      NA
## 2 E010065~      NA    NA    NA         NA    NA     NA     NA    NA      NA
## 3 E010065~      NA    NA    NA         NA    NA     NA     NA    NA      NA
## 4 E010065~      NA    NA    NA         NA    NA     NA     NA    NA      NA
## 5 E010065~      NA    NA    NA         NA    NA     NA     NA    NA      NA
## 6 E010065~      NA    NA    NA         NA    NA     NA     NA    NA      NA
## # ... with 84 more variables: beverages <int>, bicycle <int>, bookmaker <int>,
## #   books <int>, `Business Service` <int>, butcher <int>, cakes <int>,
## #   car <int>, car_repair <int>, carpet <int>, catalogue <int>, charity <int>,
## #   chemist <int>, clothes <int>, coffee <int>, computer <int>,
## #   confectionery <int>, convenience <int>, cosmetics <int>, craft <int>,
## #   deli <int>, department_store <int>, Discount <int>, doityourself <int>,
## #   dry_cleaning <int>, electrical <int>, electronics <int>, erotic <int>,
## #   estate_agent <int>, fashion <int>, florist <int>, food <int>,
## #   frozen_food <int>, funeral_directors <int>, furniture <int>, games <int>,
## #   garden_centre <int>, gift <int>, greeting_card <int>, `greetings
## #   cards` <int>, hairdresser <int>, hardware <int>, health_food <int>,
## #   hifi <int>, houseware <int>, jewelry <int>, kiosk <int>, laundry <int>,
## #   leather <int>, mall <int>, mobile_phone <int>, music <int>,
## #   musical_instrument <int>, newsagent <int>, optician <int>, outdoor <int>,
## #   pawnbroker <int>, perfumery <int>, pet <int>, photo <int>, religion <int>,
## #   salon <int>, second_hand <int>, shoes <int>, sports <int>,
## #   stationery <int>, storage_rental <int>, supermarket <int>, tailor <int>,
## #   tea <int>, ticket <int>, `ticket;convenience` <int>, toys <int>,
## #   trade <int>, travel_agency <int>, vacant <int>, variety_store <int>,
## #   video_games <int>, Vintage_Boutique <int>, watches <int>,
## #   window_blind <int>, wine <int>, yes <int>, `<NA>` <int>
```

3. A dataframe in wide format reporting:

- the LSOA code;
- the ratio of points in each class to the total LSOA area (or in other words the number of points by sqkm).

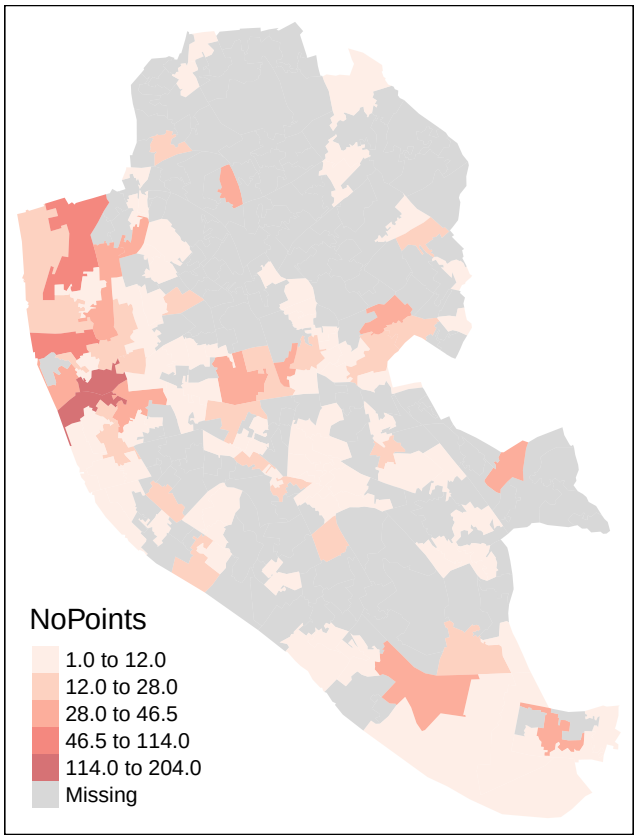
```
head(Shops_class$PointsRatioWide)
```

```
## # A tibble: 6 x 94
##   lsoa11cd alcohol  baby  Baby baby_goods   bag bakery beauty   bed betting
##   <chr>      <dbl> <dbl> <dbl>      <dbl> <dbl>  <dbl>  <dbl> <dbl>   <dbl>
## 1 E010065~      NA    NA    NA         NA    NA   5.41   1.80    NA      NA
## 2 E010065~      NA    NA    NA         NA    NA  NA     NA       NA      NA
## 3 E010065~      NA    NA    NA         NA    NA  NA     NA       NA      NA
## 4 E010065~      NA    NA    NA         NA    NA  NA     NA       NA      NA
## 5 E010065~      NA    NA    NA         NA    NA  NA     NA       NA      NA
## 6 E010065~      NA    NA    NA         NA    NA  NA     NA       NA      NA
## # ... with 84 more variables: beverages <dbl>, bicycle <dbl>, bookmaker <dbl>,
## #   books <dbl>, `Business Service` <dbl>, butcher <dbl>, cakes <dbl>,
```

```
## #   car <dbl>, car_repair <dbl>, carpet <dbl>, catalogue <dbl>, charity <dbl>,
## #   chemist <dbl>, clothes <dbl>, coffee <dbl>, computer <dbl>,
## #   confectionery <dbl>, convenience <dbl>, cosmetics <dbl>, craft <dbl>,
## #   deli <dbl>, department_store <dbl>, Discount <dbl>, doityourself <dbl>,
## #   dry_cleaning <dbl>, electrical <dbl>, electronics <dbl>, erotic <dbl>,
## #   estate_agent <dbl>, fashion <dbl>, florist <dbl>, food <dbl>,
## #   frozen_food <dbl>, funeral_directors <dbl>, furniture <dbl>, games <dbl>,
## #   garden_centre <dbl>, gift <dbl>, greeting_card <dbl>, `greetings
## #   cards` <dbl>, hairdresser <dbl>, hardware <dbl>, health_food <dbl>,
## #   hifi <dbl>, houseware <dbl>, jewelry <dbl>, kiosk <dbl>, laundry <dbl>,
## #   leather <dbl>, mall <dbl>, mobile_phone <dbl>, music <dbl>,
## #   musical_instrument <dbl>, newsagent <dbl>, optician <dbl>, outdoor <dbl>,
## #   pawnbroker <dbl>, perfumery <dbl>, pet <dbl>, photo <dbl>, religion <dbl>,
## #   salon <dbl>, second_hand <dbl>, shoes <dbl>, sports <dbl>,
## #   stationery <dbl>, storage_rental <dbl>, supermarket <dbl>, tailor <dbl>,
## #   tea <dbl>, ticket <dbl>, `ticket;convenience` <dbl>, toys <dbl>,
## #   trade <dbl>, travel_agency <dbl>, vacant <dbl>, variety_store <dbl>,
## #   video_games <dbl>, Vintage_Boutique <dbl>, watches <dbl>,
## #   window_blind <dbl>, wine <dbl>, yes <dbl>, `<NA>` <dbl>
```

Finally we can plot the results in a map, which shows the density of shops in the city of Liverpool at the LSOA level.

```r
# attach the information calculate using extRatum to the LSOA boundaries
Liv_shops_geo <- dplyr::left_join(LSOAs, Shops_total, by = "lsoa11cd")

tm_shape(Liv_shops_geo) +
  tm_fill("NoPoints", style = "fisher", palette = "Reds", alpha = 0.6)
```



Polygon data analysis

In this part, we will deal with polygon data.

We create a query to download building footprints for the city of Liverpool using OpenStreetMap data.

```
q2 <- getbb("Liverpool", limit = 100) %>%
  opq() %>%
  add_osm_feature(key = "building")

buildings <- osmdata_sf(q2)
```

We can then subset the buildings that are classified as retail.

```
retail_buildings <- subset(buildings$osm_polygons, building=="retail")
```

Then, we run the function that calculates the area in sqm covered by retail buildings in each LSOA using `areal_calc()` function.

```
Liv_retail <- areal_calc(
  polygon_layer = retail_buildings,
  higher_geo_lay = LSOAs,
  unique_id_code = 'lsoa11cd',
  crs = BNG
  )
```

The output of this function will be a dataframe containing:

- the LSOA code;
- the total area in sqkm of each LSOA;
- the area covered by the geospatial feature we have selected in each LSOA; and
- the ratio of geospatial feature area to the total LSOA area (or in other words the area covered by the geospatial feature by sqkm).

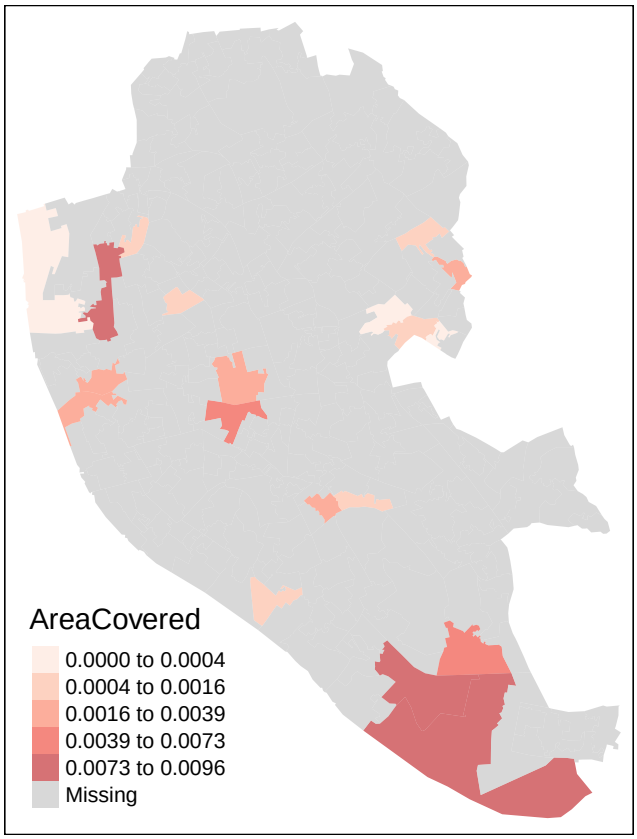
Given that everything is measured in sqkm, the ratio represents what is the % of area covered by retail buildings by sqkm. In this way, we have a relative measure that can be compared across all LSOAs and is independent of their size.

```
head(Liv_retail)
```

```
## # A tibble: 6 x 4
##   lsoa11cd  Tot_area_sqkm AreaCovered     Ratio
##   <chr>             <dbl>       <dbl>     <dbl>
## 1 E01006519         0.506  0.000946   0.00187
## 2 E01006537         1.08   0.00586    0.00544
## 3 E01006563         0.287  0.00145    0.00506
## 4 E01006570         0.443  0.00000759 0.0000171
## 5 E01006571         0.405  0.000830   0.00205
## 6 E01006572         0.136  0.000110   0.000813
```

Finally, we can plot the results, showing the total area covered by retail buildings in each LSOA in Liverpool. Note that OSM data on retail buildings are not complete for the city of Liverpool. Thus, we see too many LSOAs missing data.

```
Liv_retail_geo <- dplyr::left_join(LSOAs, Liv_retail, by = "lsoa11cd")

tm_shape(Liv_retail_geo) +
  tm_fill("AreaCovered", style = "fisher", palette = "Reds", alpha = 0.6)
```



Line data analysis

In this part, we will deal with line data.

We create a query to download highway lines for the city of Liverpool using OpenStreetMap data.

```
q3 <- getbb("Liverpool") %>%
  opq() %>%
  add_osm_feature(key = "highway")

highways <- osmdata_sf(q3)
```

We can then create subsets of the dataset such as pathways for pedestrian use.

```
pedestrian <- subset(highways$osm_lines, highway == "pedestrian")
```

Then we can calculate the total length of pedestrian pathways routes by LSOA using `line_calc()` function.

```
Liv_footways <- line_calc(
  line_layer = pedestrian,
  higher_geo_lay = LSOAs,
  unique_id_code = 'lsoa11cd',
  crs = BNG
  )
```

The output of this function will be a dataframe containing:

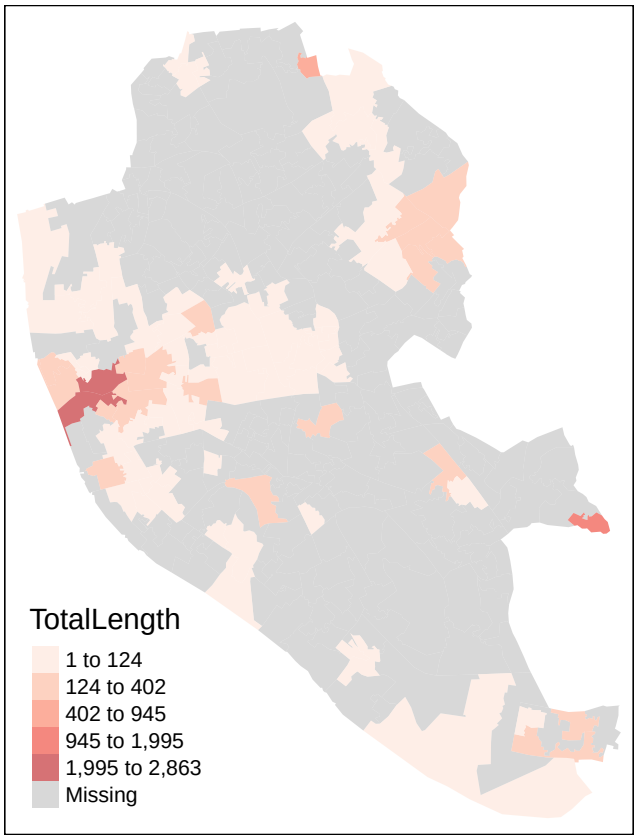
- the LSOA code;
- the total area in sqkm of each LSOA;
- the total line length by the geospatial feature we have selected in each LSOA; and
- the ratio of geospatial feature length to the total LSOA area (or in other words the length of the geospatial feature by sqkm). In this way, we have a relative measure that can be compared across all LSOAs and is independent of their size.

```
head(Liv_footways)
```

```
## # A tibble: 6 x 4
##   lsoa11cd  Tot_area_sqkm TotalLength Ratio
##   <chr>             <dbl>       <dbl> <dbl>
## 1 E01006512         0.284       22.7   79.9
## 2 E01006513         0.555      205.   369.
## 3 E01006514         0.262        9.73  37.1
## 4 E01006515         0.366       38.3  104.
## 5 E01006518         0.235       29.5  125.
## 6 E01006521         0.603       29.2   48.4
```

Finally, we can plot the results, showing the total length of pedestrian pathways in each LSOA in Liverpool. Note that the majority is around Liverpool city centre where we see darker colours.

```
Liv_footways_geo <- left_join(LSOAs, Liv_footways, by = "lsoa11cd")


tm_shape(Liv_footways_geo) +
  tm_fill("TotalLength", style = "fisher", palette = "Reds", alpha = 0.6)
```


TotalLength
1 to 124
124 to 402
402 to 945
945 to 1,995
1,995 to 2,863
Missing