CASA, University College London

aosm() – Acquiring, Filtering and Analysing OpenStreetMap Data

Geographic Information Systems and Science – Coursework

Balamurugan Soundararaj
1/7/2013

aosm() – Acquiring, Filtering and Analysing OpenStreetMap Data

INTRODUCTION

Started in 2004 as a crowd-sourced project to create an open, map of the world, OpenStreetMap has now become one of the most comprehensive, open geographic databases of the world with remarkable quality. R project for statistics computing is an open, free and powerful platform which has also proved to be an excellent tool in geographic analysis. Along with OSM data R provides a unique opportunity to create a powerful platform to access and analyse geographic data at a global level. Though there are ways to import and manipulate data from OSM currently in R (osmar) the associated methods are usually cumbersome, dealing with huge data-sets with no quick ways to finely choose and find quick statistics about the data being handled. This function is an attempt to build such a tool, combining existing methods, to quickly do such operations and get outputs in standard formats for further use.

aosm ( ) – THE FUNCTION

Syntax

name.object <- aosm ("world", "geog-filter", "tag-filter", "analysis", "type ")

Note *- Before attempting to run the function kindly read the user documentation part of the document to set up the required software and settings.*

Inputs

World – A string, which is the name of the city for which the data to be developed. The string has to be in lowercases and if the city name has spaces in it, then it has to be replaced with "-". (E.g. "london", "san-francisco")

Gfilter – A string, which is the name of the area within the city for which the data has to be extracted. The string has to be in lowercases and if the name has spaces in it, then it has to be replaced with "-". (E.g. "Islington", "city-of-london")

Tfilter – A string, denoting filter definition. The syntax is "switch_name". Where, "switch" is either "d" or "t" denoting if the name is a definition file or the tag filter in itself. A definition file is a simple text file with comma separated values where the first value is the key to be filtered and the rest of them are values within the key. A tag filter string is the same but written directly to the function without a separate file (E.g. "d_landuse" will point to a predefined text file, "t_highway,residential" will filter features with the key="highway" and value="residential"). If there is only one value then all the features with the corresponding key values are extracted.

Analysis – the type of analysis to be done on the data extracted. Currently supports the following, "default", "utm", "cn", "ar", "len".

Type – type of database into which the extracted features are converted to. Either "points", "lines" or "polygons"

Outputs

The output from the function differs significantly based on the "analysis" string in the input. The possible strings and corresponding analysis are given below.

"default" – returns an sp class object (SpatialPointsDataFrame, SpatialLinesDataFrame, SpatialPolygonsDataFrame) with the added attributes showing the key-value tags and the name tags without any CRS information

"utm" – returns a similar object to above but with the CRS information using Universal Transverse Mercator and WGS84

"cn" – return the count of features, "area" – returns the sum of areas of all the features in the resulted data in square meters and "len" – returns the length/perimeter of all the features in the resulted data in meters.

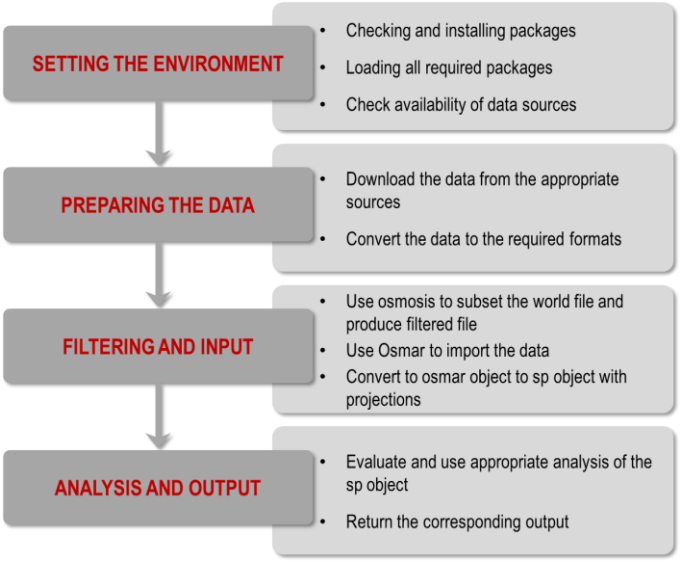
METHODOLOGY

The function aosm( ) takes 5 inputs and applies 16 sub functions on the inputs to generate the results. As a brief explanation, the function first sets the environment by installing all the required packages. It then checks the WD for boundary shape file and if found, converts it to .poly files. It then evaluates the inputs to see where and in what formats do the required data exists and creates a data frame explaining the situation. This involves checking for locally available data and data available in the internet sources for all compatible formats. The next step checks the situation and evaluates if the function can continue. If it finds any errors, it reports the error and shuts down the function before any time intensive task is started.

Once the validity of the inputs are confirmed, the function then arranges the data from the available formats, downloads it and converts it to the desired format. Here local data is given preference over data on the internet. Once the data is arranged, the function invokes osmosis for the filtering process and makes a system() call based on the inputs. Once the

filtering process is complete, osmar is used to import the filtered data file to a sp object and the extra attribute information is attached to it. The resulted file is then projected using UTM projection and WGS84 datum. As the final step based on the inputs the function applies appropriated analysis on the sp object and returns the results.

The overall methodology is explained as below,



DEMO

To demonstrate the capabilities of the function, the following code is used to produce the .png image and calculations which was used to produce the map shown in Figure 1.

```
>map<-aosm("london","city-of-london","t_building","utm","polygons")
>area<-aosm("london","city-of-london","t_building","ar","polygons")
>length<-aosm("london","city-of-london","t_building","len","polygons")
>count<-aosm("london","city-of-london","t_building","cn","polygons")
> plot(map,col="#B43104",bor="transparent")
```

POTENTIAL APPLICATIONS

The function can be used in conjecture with other tools to provide where this can be used to extract information from the OSM data in parts and combined to form a visualisation at a larger level. (E.g. a borough map of London can be made showing the % area of open space in them). The function can be used to get quick statistics for any OSM data file. (E.g. the file can be renamed into “world_gfilter_tfilter.osm” and kept in the WD to get statistics on the file). The function can also be extended to add more complex analysis functions. (E.g. functions to calculate fractal dimension of features, to automatically clean and sort the tag values).

LIMITATIONS

The major limitation of this function is its system and software requirements, which is really specific. The other limitation is the time taken to process the data which is significantly higher than operations carried out in sp objects. Apart from these all the limitations of OSM data will apply to the function as well since it depends directly on OSM for data.

REFERENCES

1. osmar, 2012. CRAN Repository. [online] Available at: http://cran.r-project.org/web/packages/osmar/index.html [Accessed 6 January 2012].
2. sp, 2012. CRAN Repository. [online] Available at: http://cran.r-project.org/web/packages/sp/index.html [Accessed 6 January 2012].
3. maptools, 2012. CRAN Repository. [online] Available at: http://cran.r-project.org/web/packages/maptools/index.html [Accessed 6 January 2012].
4. httr, 2012. CRAN Repository. [online] Available at: http://cran.r-project.org/web/packages/httr/index.html [Accessed 6 January 2012].
5. rgeos, 2012. CRAN Repository. [online] Available at: http://cran.r-project.org/web/packages/rgeos/index.html [Accessed 6 January 2012].
6. OpenStreetMap, 2012. OpenStreetMap wiki. [online] Available at: http://wiki.openstreetmap.org [Accessed 6 January 2012].
7. osmosis, 2012. OpenStreetMap wiki. [online] Available at: http://wiki.openstreetmap.org/wiki/Osmosis [Accessed 6 January 2012].
8. Haklay, M., 2010, How good is volunteered geographical information? A comparative study of OpenStreetMap and Ordnance Survey datasets, Environment and Planning B, 37(4) 682 – 703.
9. Taginfo,2012. Taginfo Database [online] Available at: http://taginfo.openstreetmap.org/ [Accessed 6 January 2012].

SYSTEM REQUIREMENTS

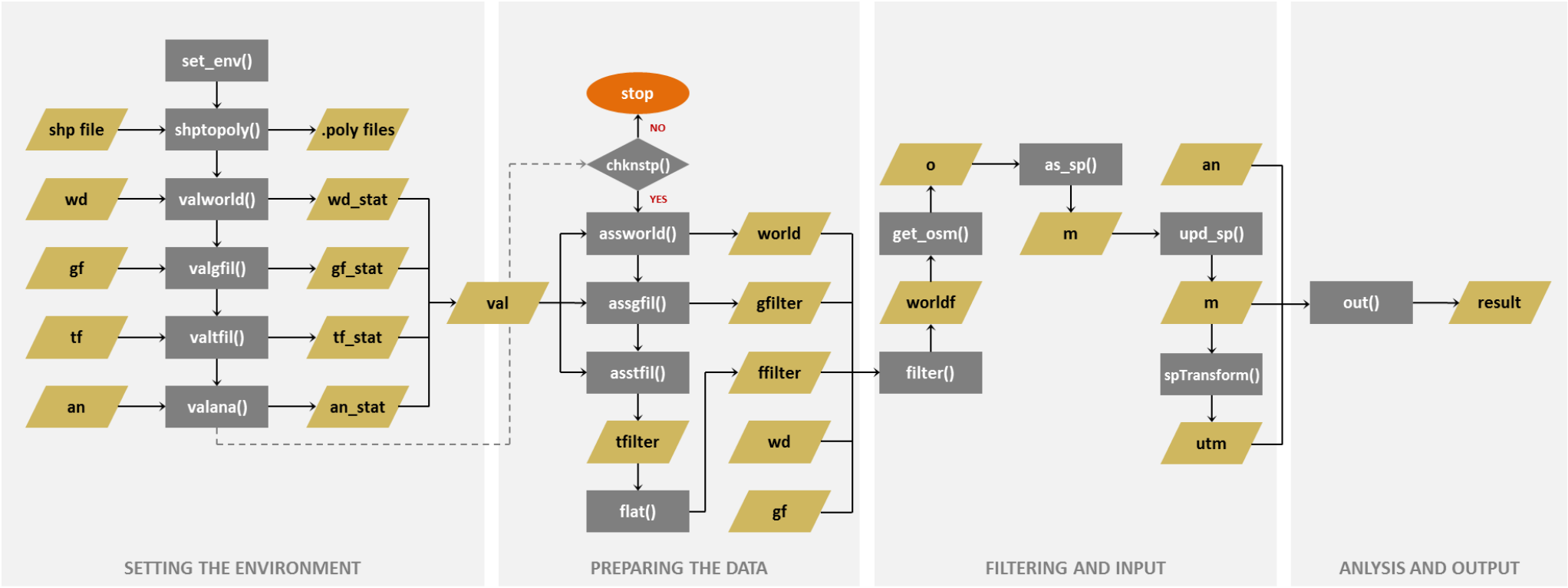
1. Windows Operating system (windows 7 preferably) with administrator rights since installation of certain software is required.
2. R 2.15.2 (http://www.r-project.org/) installed in the system.
3. Osmosis (http://wiki.openstreetmap.org/wiki/Osmosis) installed in the system with ‘osmosis.bat’ file location added to the system path variable
4. 7zip (http://www.7-zip.org/) installed in the system with ‘7z.exe’ file location added to the system path variable
5. Since there is a lot of data which needs to be downloaded and extracted it is recommended to have atleast 2 GB of free space in the hard drive
6. The function requires a working internet connection to work.

DATA AVAILABILITY

1) OSM data - http://metro.teczno.com/
2) Polygon Files - http://balaspa.50webs.com/poly/
3) Tag Definitions -<http://balaspa.50webs.com/def/>

DETAILED WORKING:

The detailed working flow diagram of all the sub functions and their corresponding inputs and outputs, along with detailed explanation of their roles are shown in the diagram below.



FLOW CHART SHOWING THE FUNCTIONING OF AOSM()

FUNCTION OBJECT ACTION

set_env ( ) – Sets the environment by checking the availability of packages, installing missing packages and loading them to the library.

shptopoly ( ) – Checks for shape file named 'boundary.shp', with 'NAME' attribute data, in the working directory and produces 'name.poly' file for each polygon in the file.

valworld ( ), valgfil ( ), valtfil ( ), valana ( ) – Takes the strings from the parent as inputs, checks for the availability of corresponding data locally and in in the internet and returns data.frame with results.

chknstp ( ) – takes the combined 'val' data.frame and checks if the input is sufficient for further action. Stops and prints appropriate error message when there is a problem.

assworld ( ), assgfil ( ), asstfil ( ) – takes in the 'val' and decides which source to use and downloads and prepares them to usable formats and gives their names as outputs.

flat ( ) – converts the 'tfilter' object into a flat string for usage with osmosis and produces 'ffilter'

filter ( ) – takes all the arguments prepared so far as inputs and calls osmosis through system( ) and produces the filters world file – 'worldf'

get_osm ( ) – takes the worldf file and converts it to an osmar object using osmar package resulting in an object 'o'

as_sp ( ) – Takes the osmar object and converts it into sp based on the type specified in the parent function resulting in an sp object – 'm', which has no CRS information and no attribute data

upd_sp ( ) – adds the key-value information to the sp object m as attributes. The attributes added are 'the id-key-value' for tags filtered (id_tf, k_tf, v_tf) and the 'name' information (id_nm, k_nm, v_nm).

spTransform ( ) – Takes the sp object m and adds CRS information to it. The projection system added is 'Universal Traverse Mercator' with the datum 'WGS 84'

out ( ) – Calculates and returns appropriate output for parent based on the 'analysis' value.

Total Area of Borough	**2.59 km^2**	No of Buildings	**1364**
Total Built up Area	**1.31 km^2**	Total Perimeter Length	**337.4 km**
Ground Coverage	**50.57%**	Source: OpenStreetMap	

Figure 1 - Demonstration of Result from the Function.