Paper 3102-2019

Spatial Analysis of Humanitarian OpenStreetMap Team data using SAS ODS Graphics Procedures

Michael Matthews, Innotwist Pty Ltd

ABSTRACT

The Humanitarian OpenStreetMap Team (HOT) community consists of volunteers from around the globe. HOT tasks involve developing maps that identify communities and infrastructures based on satellite imagery. These maps are then used to assist aid organizations such as the Red Cross during humanitarian crises and for general community development in areas that are often not covered by the mapping products that most of us take for granted. This presentation examines OpenStreetMap data and introduces HOT and some of the associated mapping tasks, including assisting the aid efforts during the May 2018 Ebola outbreak in the Democratic Republic of the Congo. Analysis is performed on the data using the SAS® ODS Graphics procedures (including PROC SGMAP) to visualize the contributions to OpenStreetMap both spatially and over time.

INTRODUCTION

According to the United Nations, the number of people affected by a humanitarian crisis has doubled in the past decade. Open source technology is increasingly being used in a number of applications to create new solutions to allow aid organizations to respond to disasters such as earthquakes, floods, tsunamis or infectious disease outbreaks. Particularly in the developing world, many communities have not been included in traditional mapping products. This presents a challenge for humanitarian organizations responding to disasters, whose volunteers may at best be given a hand-drawn, basic map to reach isolated and vulnerable communities.

The Humanitarian OpenStreetMap Team (HOT) is changing this access and inclusion by bringing together a community of volunteers from around the world (over 100,000 people from more than 60 countries) who are placing previously untracked communities on the map through collaborative geospatial data collection. Organizations such as the Red Cross, Médecins Sans Frontières, USAID and local government organizations post projects through the HOT web interface for mappers to work on.

Using satellite imagery, volunteers are allocated specific grids within a geographic region to trace infrastructure such as buildings, roads, rail or rivers. These map edits are validated and pushed live into OpenStreetMap so that on-the-ground responders can access the maps on their mobile devices as they move into the crisis areas. The mapping data is also being used for communities' evidence-based decisions, predictive analysis and preparation for both natural and medical disasters.

Global contributors to OpenStreetMap result in a rich source of data for analysis that can be used for many purposes. This paper analyses OpenStreetMap data and some of the Humanitarian OpenStreetMap Team mapping tasks, including a review of the aid efforts during the Ebola outbreaks in the Democratic Republic of the Congo during 2018. Analysis is performed on the data using the SAS® ODS Graphics procedure PROC SGMAP to visualize the contributions to OpenStreetMap both spatially and over time. This type of analysis is useful to review volunteer contributions and identifies mapping 'hot spots' during particular crisis events to track which areas receive significant volume of mapping updates.

OPENSTREETMAP®

OpenStreetMap® is a map of the world built by a community of mappers who donate their time and skills to create mapping data available for everyone.

Consumers of OpenStreetMap, and other commercial mapping solutions, often consider the visual display of maps to be the final product, however with OpenStreetMap the data itself is the primary asset, with the rendering of the maps only one of the many possible uses.

The OpenStreetMap data is open; licensed under the Open Data Commons Open Database License (ODbL) by the OpenStreetMap Foundation (OSMF).

The default OpenStreetMap presentation at https://www.openstreetmap.org is shown in Figure 1:

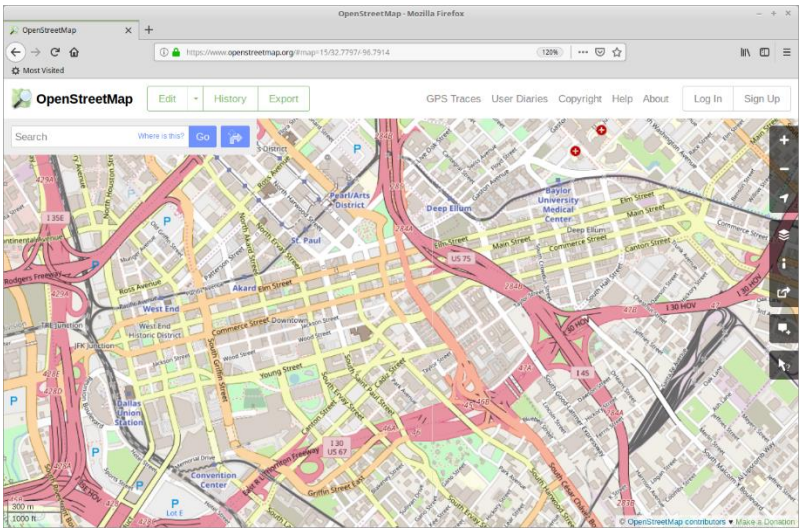


Figure 1. Standard OpenStreetMap layer

The underlying OpenStreetMap data supports producing many displays with a different emphasis; for example, Figure 2 shows a transport layer including the train and other public transport routes as the primary focus:

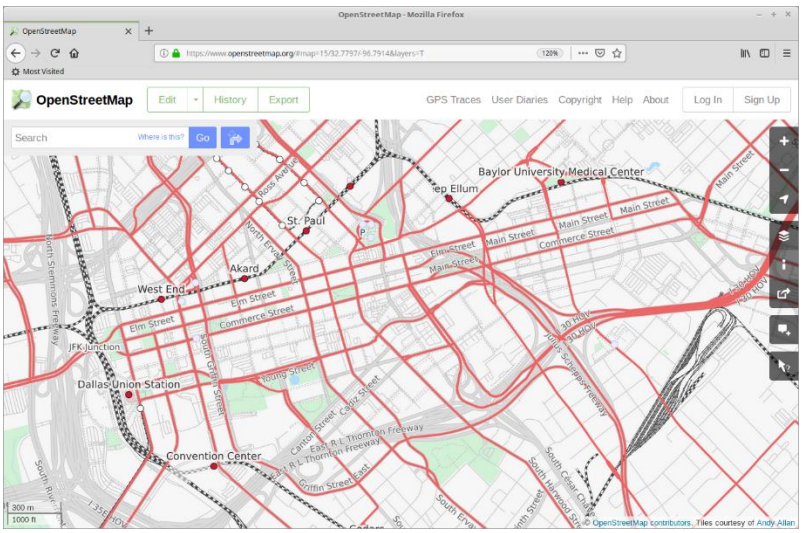


Figure 2. Transport Map OpenStreetMap layer

The generation of the maps shown on the OpenStreetMap website are rendered using a Tile Server which generates a group of image tiles that are then seamlessly joined in the browser to create the complete map display. Different maps can be rendered based on the specific requirements; along with the standard view, there are transport maps (that highlight train, bus and ferry routes), humanitarian views (that highlight medical facilities, water sources, and other important infrastructure), cycling maps and sea maps for navigation as a few examples.

The OpenStreetMap data is available to copy and use for any purpose within the license agreement.

The data is available from numerous sources, including:

1. APIs: OpenStreetMap and Overpass API are two examples of many available. The licenses for these sources limit usage and should be reviewed before use.
2. Database copy: Planet OSM enables you to copy the entire global database (100+ GB). Other providers offer useful regional extracts at different country and/or state levels. It is also possible with a full copy to create a process where regular deltas are applied to keep a local copy of the OpenStreetMap database current without having to download the entire globe on a regular basis.
3. Tile Servers: The OpenStreetMap tile servers and others are available for personal use, however they are often not licensed for commercial use due to their limited funding and infrastructure. It is possible to create your own tile server, such as the way SAS Institute have done, to support the OpenStreetMap layers in their mapping solutions.

DATA CONCEPTS

The following fundamental concepts regarding the data used by OpenStreetMap are required to understand the analysis performed in this paper:

Node: A point which may be standalone as a marker to identify an item or place, or part of a way.

Way: An ordered collection of Nodes that make up a larger object. This may be a street, building, lake, park, or any other object represented as either a line or polygon.

Relation: A collection of related Nodes and Ways. An example might be multiple buildings that are related as a single hospital.

Tag: A descriptive attribute that can be added to Nodes, Ways or Relations. This might be a street name, building height, road surface etc. The tags provide rich data for both the tile rendering and for analysis.

Changeset: A batch of nodes/ways/relations/tags provided by a contributor to the OpenStreetMap project through one of the supported editors. The underlying database is not updated by users with every minor change made; only at the end of an editing session where the changes are saved as a changeset.

HOTELS IN DALLAS

The rich data in the OpenStreetMap database includes many tags describing the nodes, ways and relations. Many of these are not visible in the standard map displays, however the data is able to be used to perform analysis and reporting.

To introduce the use of OpenStreetMap data with SAS Software, this example uses the OverPass API to extract a small volume of points of interest (hotels) around the venue of

the 2019 SAS Global Forum. The data is then processed using BASE SAS and the SAS procedure SGMAP.

The following text file is using the Overpass API request format selecting the "Dallas" area, and then searching for nodes, ways and relations with the tag tourism="hotel", along with the "Kay Bailey Hutchinson Convention Center" which is the location for the SAS Global Forum 2019:

```
[out:json][timeout:25];
(area[name="Dallas"];)->.searchArea;
(
  node["tourism"="hotel"](area.searchArea);
);
out body;
(
  way["tourism"="hotel"](area.searchArea);
  rel["tourism"="hotel"](area.searchArea);
  rel["name"="Kay Bailey Hutchinson Convention Center"](area.searchArea);
);
out center;
```

The following code uses the SAS procedure HTTP to call the Overpass API, with the specifications provided above in the file dallas_hotels_spec.txt. The results of the API call are written in JSON format to the file dallas_hotels.json:

```
filename spec "dallas_hotels_spec.txt";
filename resp "dallas_hotels.json";

proc http url="http://overpass-api.de/api/interpreter"
          in=spec out=resp
          method="post"
          ct="application/x-www-form-urlencoded";
run;
```

Note that it is worthwhile saving the results of any requests rather than calling the API repeatedly if the data is unlikely to change. This will improve your program performance and prevent unnecessary calls to the API servers.

The JSON libname engine is then used to read dallas_hotels.json into a SAS dataset:

```
libname hotels JSON fileref=resp;

data work.hotel_details(drop=ordinal_elements);
  merge hotels.elements(keep=id ordinal_elements lat lon)
        hotels.elements_tags(keep=ordinal_elements name);
  by ordinal_elements;
  if name="Kay Bailey Hutchinson Convention Center" then
    type="SASGF";
  else
    type="Hotel";
run;
```

The resulting SAS dataset now contains a list of hotels as shown in Output 1:

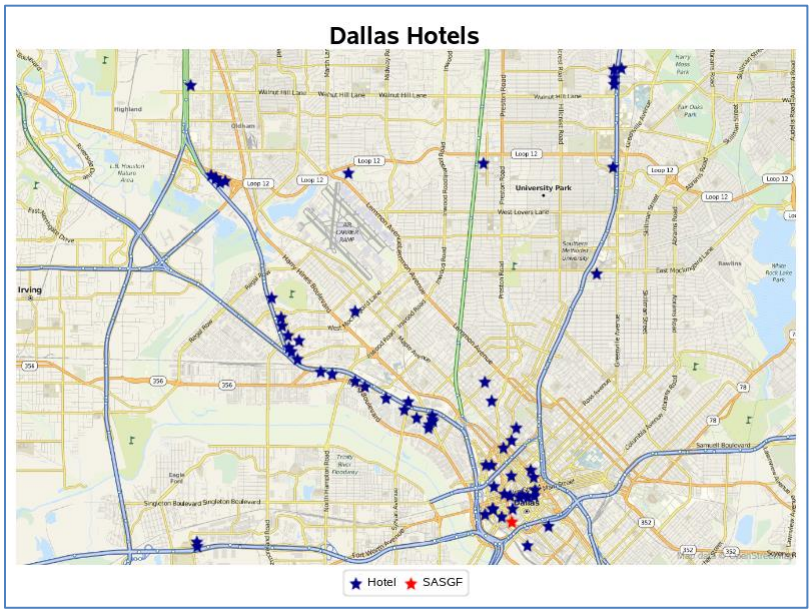
Obs	id	lat	lon	name	type
1	1125698005	32.8362	-96.7777	The Highland Dallas	Hotel
2	3158674753	32.8039	-96.8320	Renaissance Dallas Hotel	Hotel
3	3334716952	32.7689	-96.8948	Hampton Inn & Suites	Hotel
4	3334716962	32.7679	-96.8947	Comfort Suites	Hotel
5	4135373421	32.9240	-96.8171	Hilton Dallas Lincoln Centre	Hotel

Output 1. List of Hotels in Dallas

The lat and lon variables are now in a form suitable for presenting visually on a map. The SAS procedure SGMAP is used to generate the map using BASE SAS:

```
proc sgmap plotdata=work.hotel_details;
  openstreetmap;
  title h=2 "Dallas Hotels";
  scatter x=lon y=lat / name="hotels" group=type
                        markerattrs=(symbol=starfilled size=15px);
  keylegend "hotels" / title="";
run;
```

The resulting image from the PROC SGMAP code is shown in Output 2:



Output 2. Hotels in Dallas

The OPENSTREETMAP statement supplied to PROC SGMAP ensures that the rendered background for the map is using the SAS maintained OpenStreetMap tile server.

The GROUP option on the SCATTER statement provides the different appearance for the hotels and location of the SAS Global Forum including the legend.

This same technique for extracting OpenStreetMap data can be used for other items of interest, such as hospitals, golf courses, mines, parks, train stations etc., which can be useful for other analytics and visualizations.

OPENSTREETMAP CHANGESETS

The OpenStreetMap database is updated using various editing programs. These editors do not write individual nodes and ways to the database for every altered item; instead the process of saving the results of an edit session creates a batch of changes called a "changeset".

OpenStreetMap data is often available in XML format when using the various open source tools to extract the data. An example of this data is the list of the 100 most recent changesets for a contributor to OpenStreetMap which is available from the OpenStreetMap API.

The following PROC HTTP request will return the changeset data (for the contributor supplied in the SAS Macro variable &name) in XML format:

```
filename resp "user_changesets.xml";

proc http
url="https://api.openstreetmap.org/api/0.6/changesets?display_name=&name"
          out=resp method="get";
run;
```

BASE SAS can read the generated XML files when provided with a suitable XMLMAP for the schema. The XMLMAP used here for the OpenStreetMap XML format is available at https://github.com/m-matthews/hot-sas/blob/master/data/osm_map.xml.

The following code will read the XML file into SAS dataset format:

```
filename osm_map "osm_map.xml";
libname osm_cs xmlv2 "user_changesets.xml" xmlmap=osm_map;

data work.osm_changesets(keep=id created_at changes_count comment lat lon);
  merge osm_cs.changesets
        osm_cs.changeset_tags(where=(key="comment")
                              rename=(value=comment));
  by id;
  lat = mean(min_lat, max_lat);
  lon = mean(min_lon, max_lon);
run;
```

The output will have a similar appearance to Output 3:

Obs	id	created_at	changes_count	comment	lat	lon
1	64928732	27NOV18:09:37:37	5	#hotosm-project-5493 Tanzania Development Trust, Kiteto missing buildings	-5.64084	36.588
2	64795164	22NOV18:22:01:51	276	#hotosm-project-5507 #Ebola2018 #OSM-CD	0.12989	29.308
3	65257872	07DEC18:05:14:24	20	#hotosm-project-5509 #MapPH #UPNOAH #FoodSecurityPH #OSMPH #YouthMappers	7.96546	125.447
4	65509973	15DEC18:21:30:24	11	#hotosm-project-5596 #missingmaps #GPSDD	3.35548	32.740
5	65541240	17DEC18:07:54:21	123	#hotosm-project-5601 #osm_cd #ebola2018	0.11938	29.343
6	66265509	13JAN19:04:37:22	30	#hotosm-project-5660 #osm_cd #ebola2018 source=DigitalGlobe Premium	1.09103	29.644

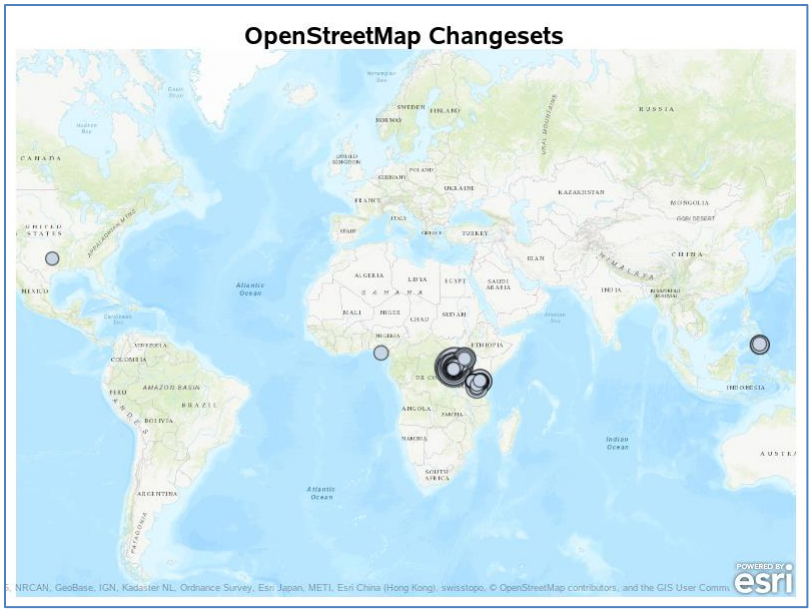
Output 3. List of Changesets

The changeset data can then be represented using PROC SGMAP and the ESRIMAP statement to provide an alternative background to the OpenStreetMap tile server:

```
%let esri_url=http://services.arcgisonline.com/arcgis/rest/services;

proc sgmap plotdata=work.osm_changesets;
  esrimap url="&esri_url/World_Topo_Map";
  bubble x=lon y=lat size=changes_count;
run;
```

The resulting image in Output 4 shows the location of the changeset centroids on a topographical map of the globe:



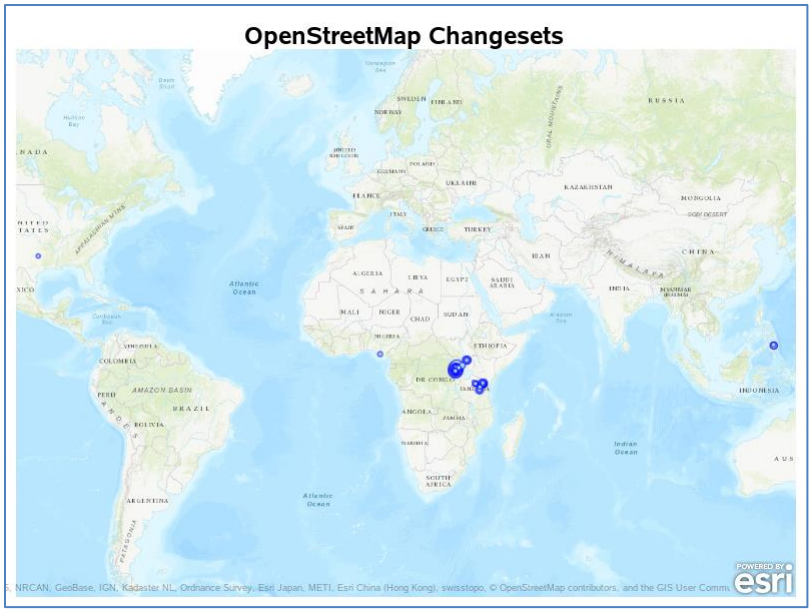
Output 4. OpenStreetMap Changesets

Additional options can be supplied to change the colors and increase the visibility where the bubbles are overlaid. The following example code includes the SAS procedure TEMPLATE to control the colors and the SGMAP BUBBLE statement options BRADIUSMIN and BRADIUSMAX to control the size of the bubbles:

```
proc template;
  define style styles.mystyle;
    parent=styles.htmlblue;
    style GraphDataDefault from GraphDataDefault /
          color=lightblue;
    style GraphOutlines from GraphOutlines /
          contrastcolor=blue linestyle=1;
  end;
run;

proc sgmap plotdata=work.osm_changesets;
  esrimap url="&esri_url/World_Topo_Map";
  bubble x=lon y=lat size=changes_count /
         bradiusmin=2px bradiusmax=8px;
run;
```

The resulting image from PROC SGMAP is shown in Output 5:



Output 5. OpenStreetMap Changesets

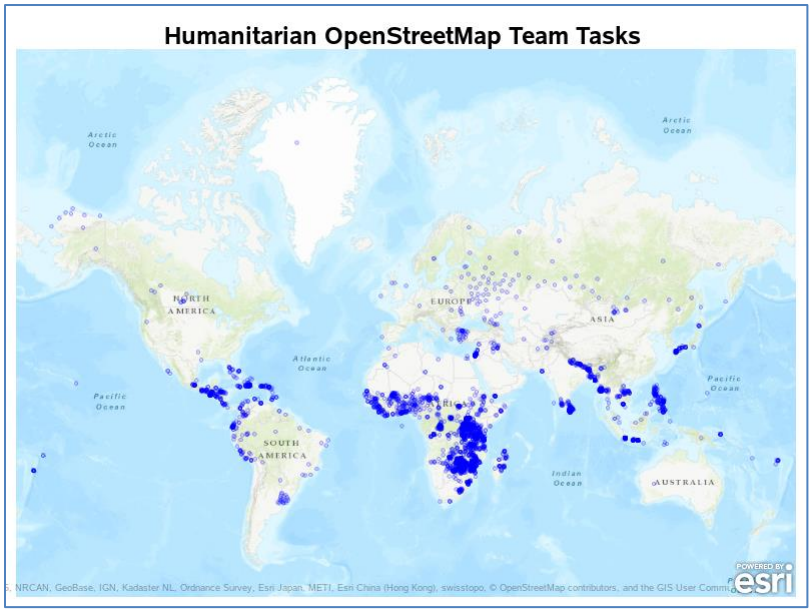
HUMANITARIAN OPENSTREETMAP TEAM

The Humanitarian OpenStreetMap Team (HOT) is dedicated to humanitarian action and community development through open mapping. The HOT community is made up of volunteers, local community leaders, and professionals who work together to provide map data which revolutionises disaster management and reduces risks for vulnerable communities. Many organizations request projects through the HOTOSM website, including the Red Cross, Médecins Sans Frontières, USAID and local government organizations.

The following PROC SGMAP code is used to display the location of the Humanitarian OpenStreetMap Team global tasks:

```
proc sgmap plotdata=work.hotosm_summary
           noautolegend;
  esrimap url="&esri_url/World_Topo_Map";
  title h=2 "Humanitarian OpenStreetMap Team Tasks";
  scatter x=lon y=lat / markerattrs=(symbol=circle color=blue size=4px);
run;
```

The display of global Humanitarian OpenStreetMap Team tasks is shown in Output 6:

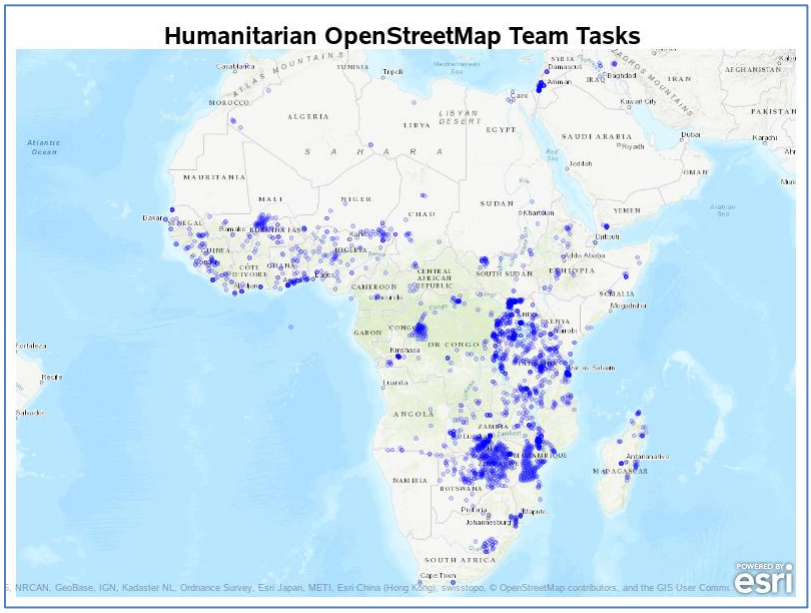


Output 6. Humanitarian OpenStreetMap Team global tasks

It is possible to change the region displayed with PROC SGMAP by adding a WHERE clause to subset the PLOTDATA on the lat and lon variables to view a segment of the globe:

```
proc sgmap plotdata=work.hotosm_summary
           noautolegend;
  where lat between -35 and 35 and lon between -29 and 57;
  esrimap url="&esri_url/World_Topo_Map";
  title h=2 "Humanitarian OpenStreetMap Team Tasks";
  scatter x=lon y=lat / markerattrs=(symbol=circle color=blue size=4px);
run;
```

The resulting display in Output 7 shows Humanitarian OpenStreetMap Team tasks in Africa:



Output 7. Humanitarian OpenStreetMap Team tasks in Africa

DEMOCRATIC REPUBLIC OF THE CONGO EBOLA CRISIS

2018 saw two major outbreaks of the highly infectious Ebola virus in separate regions of the Democratic Republic of the Congo, which was declared an international health emergency by the World Health Organization. By November, the later outbreak had become the second biggest Ebola outbreak in recorded history with over 1,000 recorded cases of infection.

The Humanitarian OpenStreetMap Team conducted numerous projects in the Democratic Republic of the Congo during these crises with multiple requests for volunteers from around the globe to identify and edit vast tracts of unmapped terrain and update population centers.

This section performs an analysis of the changes to the OpenStreetMap database over the period of the two Ebola crises. The data is extracted from a regional subset of the Planet OSM data available at the Geofabrik website http://download.geofabrik.de/africa/congo-democratic-republic.html.

The data can be converted from the supplied OSM PBF format into a CSV file to be easily consumed using SAS Software by using one of the many OpenStreetMap tools available. The following use of OSMCONVERT extracts all Nodes, and the timestamp for their last change, along with the latitude and longitude:

```
osmconvert congo-democratic-republic-latest.osm.pbf \
           --drop-ways --drop-relations \
           --csv="@id @timestamp @lat @lon" --csv-headline \
           -o=drc-nodes.csv
```

The resulting CSV file can be imported with the SAS procedure IMPORT:

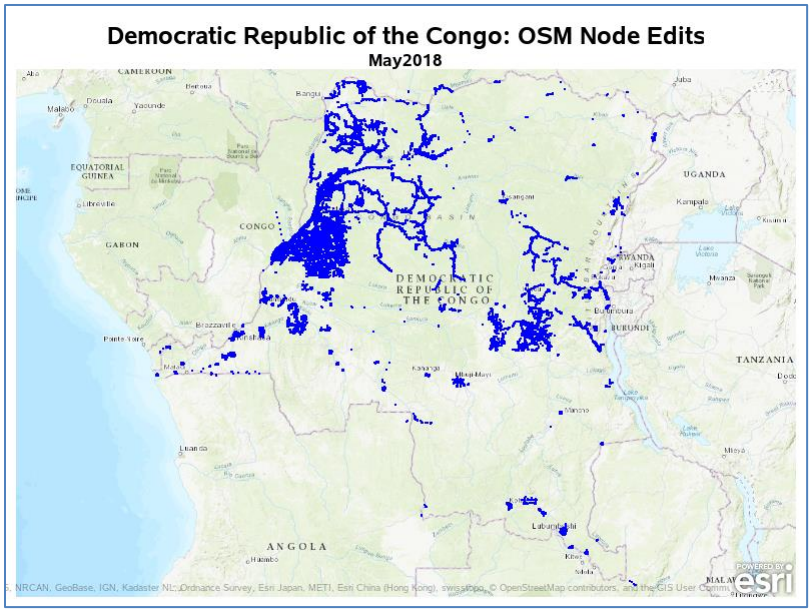
```
proc import datafile="drc-nodes.csv"
            out=work.drc_nodes
            dbms=dlm
            replace;
  delimiter="09"x;
run;

data work.nodes(keep=lat lon month week);
  set work.drc_nodes;
  month=intnx("MONTH", datepart(timestamp), 0);
  week=intnx("WEEK", datepart(timestamp), 0);
  format month monyy7. week yymmddd10.;
run;
```

This data can then be visualized with PROC SGMAP, in this case for the month of May 2018:

```
proc sgmap plotdata=work.nodes(where=(month="01may2018"d))
           noautolegend;
  esrimap url="&esri_url/World_Topo_Map";
  title h=2 "Democratic Republic of the Congo: OSM Node Edits";
  title2 h=1.5 "May2018";
  scatter x=lon y=lat / markerattrs=(symbol=circle color=blue size=2px);
run;
```

The resulting scatter plot is shown in Output 8:



Output 8. OpenStreetMap node edits for May 2018

The scatter plot displayed shows clearly the OpenStreetMap edits occurring in the western side of the country impacted by the Ebola crisis during May 2018. Unfortunately, a scatter plot can hide the actual number of points present when a solid mass as shown in Output 8 exists. It is not possible to visually determine if a single blue pixel represents a single node, or 10,000 node edits.

One solution to the issue of overlapping points in a scatter plot is to use a binning technique, where the area is divided (often into a square based grid) and then counts within those areas are used to generate colored regions of intensity.

The default 'square' grid technique is known to have potential bias and there are many discussions on the benefits of using the alternative hexagonal based binning technique.

The SAS/STAT® software procedure SURVEYREG can produce the required hexagonal regions and statistics to generate this style of map display:

```
ods select none;
ods output fitplot=work.hexmap;
proc surveyreg data=work.nodes
               plots(nbins=70 weight=heatmap)=fit(shape=hex);
  where month="01May2018"d;
  model lat=lon;
run;
ods select all;
```

A subset of the PROC SURVEYREG output is displayed in Output 9:

Obs	hID	XVar	YVar	WVar
1	1	30.2410	-13.4406	30
2	1	30.4368	-13.5609	30
3	1	30.4368	-13.8014	30
4	1	30.2410	-13.9217	30
5	1	30.0451	-13.8014	30
6	1	30.0451	-13.5609	30
7	2	30.0451	-13.0798	13
8	2	30.2410	-13.2001	13
9	2	30.2410	-13.4406	13
10	2	30.0451	-13.5609	13
11	2	29.8492	-13.4406	13
12	2	29.8492	-13.2001	13

Output 9. PROC SURVEYREG output for hexagonal binning

There are 6 observations per hID; the XVar and YVar are the boundary coordinates for the hexagon, while the WVar is the number of observations (node edits) present within this boundary.

Using the SAS procedure UNIVARIATE on the variable WVar produces useful percentiles to create visual banding ranges. Based on the current data the following is used:

```
proc format;
  value hexgrp 1-4='A'
               5-9='B'
               10-49='C'
               50-199='D'
               200-499='E'
               500-1999='F'
               2000-6999='G'
               7000-high='H';
  value $hexgrp 'A'='1-4'
                'B'='5-9'
                'C'='10-49'
                'D'='50-199'
                'E'='200-499'
                'F'='500-1999'
                'G'='2000-6999'
                'H'='7000+';
run;
```

The dataset created by PROC SURVEYREG is then converted into two datasets; one describing the boundaries (work.hexmap_map) and the other the response (work.hexmap_resp):

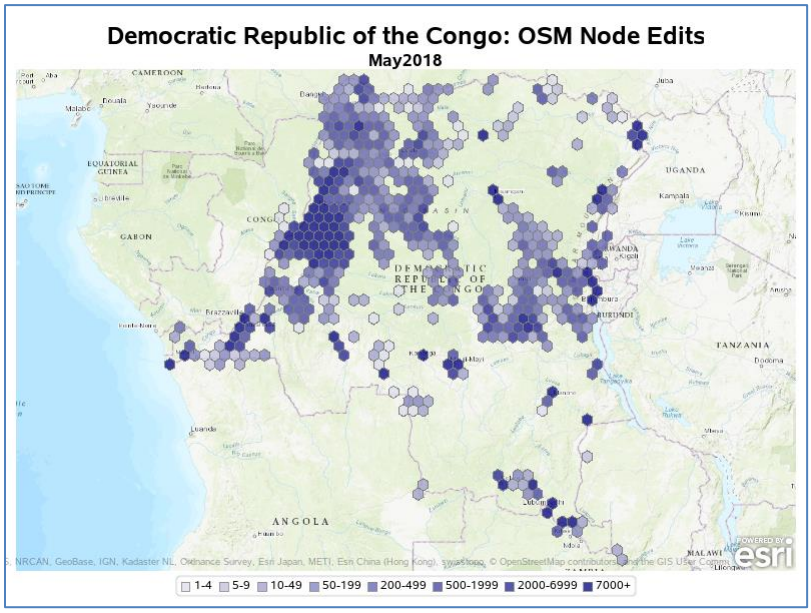
```
data work.hexmap_map(keep=id x y)
     work.hexmap_resp(keep=id wvar gvar);
  set work.hexmap(rename=(hid=id xvar=x yvar=y));
  where id ne .;
  by id;
  output work.hexmap_map;
  if last.id;
  gvar=put(wvar,hexgrp.);
  format gvar $hexgrp.;
  output work.hexmap_resp;
```

```
run;
```

These two datasets can then be used by PROC SGMAP to create the hexagons and display the density of node changes in each defined area by using a CHOROMAP statement:

```
proc sgmap mapdata=work.hexmap_map
           maprespdata=work.hexmap_resp;
  esrimap url="&esri_url/World_Topo_Map";
  title h=2 "Democratic Republic of the Congo: OSM Node Edits";
  title2 h=1.5 "May2018";
  choromap gvar / name="hexes" lineattrs=(color=gray) transparency=0.25;
  keylegend "hexes" "Nodes";
run;
```

The output from the PROC SGMAP using the hexagonal binning is shown in Output 10:



Output 10. OpenStreetMap node edits for May 2018 using hexagonal binning

This shows clearly the intense level of node edits in the western area of the Democratic Republic of the Congo. There are also other areas with high edit levels, however these may be due to the presence of population centers, as OpenStreetMap edits are always occurring globally outside of disaster response efforts.

It is possible to check the presence of population centers by adding cities and towns to the display. The extraction of this data can be achieved with open source OpenStreetMap tools. In the following script OSMOSIS can be used to extract nodes with tags of place="city" and place="town" from the current extract of the country:

```
osmosis --read-pbf file=congo-democratic-republic-latest.osm.pbf \
        --nkv keyValueList="place.city,place.town" \
        --write-xml drc-city-town.osm
```

Note that it would be possible to use OSMCONVERT on the resulting drc-city-town.osm file to convert it into a CSV format, however the previously discussed XMLMAP file can be used to read it directly into SAS:

```
filename osm_map "osm_map.xml";
libname osm_ct xmlv2 "drc-city-town.osm"
                     xmlmap=osm_map;

data work.citytown(keep=id lat lon name place);
```

```
  merge osm_ct.nodes
        osm_ct.node_tags(where=(key in ("name", "place")));
  by id;
  retain name place;
  if key="name" then name=value;
  else if key="place" then place=value;
  if last.id;
run;
```

The resulting SAS dataset shown in Output 11 contains one record for each node:

Obs	id	lat	lon	name	place
1	27043346	-4.32171	15.3126	Kinshasa	city
2	27043347	-4.31683	15.3269	Barumbu	town
3	27043358	-4.32865	15.3020	Lingwala	town
4	27043362	-4.33955	15.2662	Kintambo	town
5	27043390	-4.34647	15.3049	Kasa-Vubu	town

Output 11. List of cities and towns in the Democratic Republic of the Congo

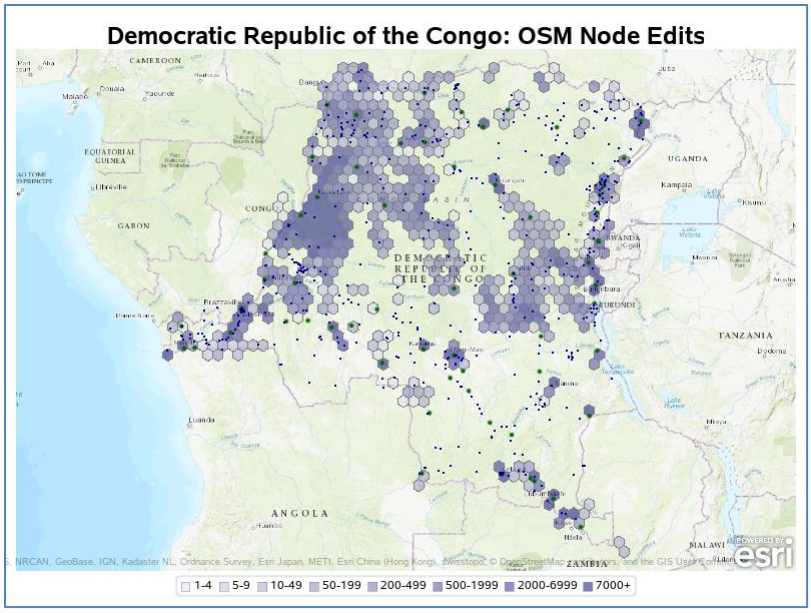
This dataset can then be converted into a format suitable for PROC SGMAP SCATTER statements by creating two distinct sets of variables for the lat and lon of the cities and towns:

```
data work.citytown(keep=lat_c lon_c lat_t lon_t place name);
  set work.citytown;
  if place="city" then do;
    lat_c=lat;
    lon_c=lon;
  end;
  else do;
    lat_t=lat;
    lon_t=lon;
  end;
run;
```

The following PROC SGMAP code then adds the city and town points on top of the existing hexagonal binning map (including transparency), using two SCATTER statements to perform this action:

```
proc sgmap mapdata=work.hexmap_map
           maprespdata=work.hexmap_resp
           plotdata=work.citytown;
  esrimap url="&esri_url/World_Topo_Map";
  title h=2 "Democratic Republic of the Congo: OSM Node Edits";
  choromap gvar / name="hexes" lineattrs=(color=gray) transparency=0.5;
  scatter x=lon_c y=lat_c / markerattrs=(symbol=circlefilled
                                         color=darkgreen size=5px);
  scatter x=lon_t y=lat_t / markerattrs=(symbol=circlefilled
                                         color=darkblue size=2px);
  keylegend "hexes" "Nodes";
run;
```

Note that using the two SCATTER statements allows direct control of the marker attributes. The resulting image follows in Output 12:



Output 12. OpenStreetMap node edits for May 2018 including city and town overlay

The visualization clearly shows that the dense node edits outside of the Ebola crisis zone are mostly related to population centers and expected to occur frequently with general ongoing OpenStreetMap editing. The area of the Ebola crisis zone also shows that it is not the site of major cities and towns, and yet is the subject of extensive edits most likely related to the Humanitarian OpenStreetMap Team projects.

ANIMATION

Map images are a powerful way of presenting data, however the OpenStreetMap edits are performed over time. The time dimension can be shown by animating a series of these images.

The code used in the preceding section to create the hexagons can be used in a macro loop to generate the sequence of images. Unchanged, however, this will result in a very unstable set of images, as the SGMAP procedure will constantly center the image based on the current data projected onto the map. To keep a consistent background image, a method needs to be included to keep the area constant. This is best performed by adding two points (top left / bottom right) that are outside of the existing data area and including these in a separate SCATTER statement. This will ensure that the images are consistent in area:

```
proc summary data=work.nodes nway;
  var lat lon;
  output out=work.summary(drop=_type_ _freq_)
         min(lat lon)=min_lat min_lon
         max(lat lon)=max_lat max_lon;
run;

data work.plot_limits(keep=x y);
  set work.summary;
  x=min_lon-1; y=min_lat-1; output;
  x=max_lon+1; y=max_lat+1; output;
run;
```

The SAS dataset work.plot_limits now contains a set of suitable points to create a consistent graph area. To prevent this from adding unwanted points on the output image, it can be

added to PROC SGMAP with a SCATTER statement using TRANSPARENCY=1.0. These points will then be invisible on the generated image, yet still control the image display area:

```
proc sgmap mapdata=work.hexmap_map(where=(month="01&month"d))
           maprespdata=work.hexmap_resp(where=(month="01&month"d))
           plotdata=work.plot_limits;
  esrimap url="&esri_url/World_Topo_Map";
  title h=2 "Democratic Republic of the Congo: OSM Node Edits";
  title2 h=1.5 "&month";
  scatter x=x y=y / transparency=1.0;
  choromap gvar / name="hexes" lineattrs=(color=gray) transparency=0.25;
  keylegend "hexes" "Nodes";
run;
```

BASE SAS Software includes the ability to animate a series of images into GIF and SVG files, however due to the SGMAP procedure including bitmaps from the underlying tile server, the smooth fading between frames available to many other Statistical Graphics (SG) procedures is not available in the current release of SAS Software. An alternative method of producing the animation outside of the SAS environment is using the open source tool Imagemagick® to render the individual frames into a smooth animation.

The following Imagemagick command will convert all the files named 04_drc_hexmonth*.png into 04_drc_hexmonth.mov with a specified delay between frames and importantly a morph factor which specifies the number of intermediate fading frames for smooth animation:

```
convert -delay 10 -morph 12 04_drc_hexmonth*.png -loop 0 miff:- | convert -
04_drc_hexmonth.mov
```

The resulting animation can be viewed on YouTube at https://youtu.be/brvmSHEmhWU, which shows clearly the volume of edits in the west of the country during the April - July 2018 outbreak, and then in the north east near the border with Uganda and Rwanda starting in August 2018.

NODE LEVEL ANIMATION

The hexagonal binning process allows for an understanding of the density of the node edits occurring over the given time periods, however there is still value in visualizing the node edits while potentially ignoring the density.

The limited resolution could be overcome by zooming into specific regions of interest, however in this example the full map of the Democratic Republic of the Congo is used.

The previous animation used the morph option to merge smoothly between frames, however additional smoothing is generated within the images in this example. To perform this, four sets of node lat/lon pairs are created. One pair for the current week (w1_lat and w1_lon), and others for each preceding week:

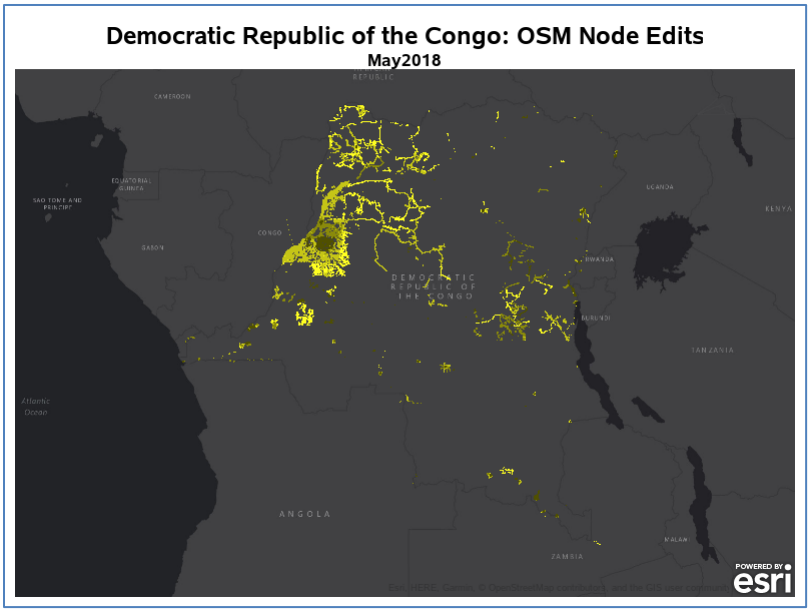
```
data work.plotdata(keep=w1: w2: w3: w4: x y);
  set work.nodes
      work.plot_limits(in=lim);
  format week;
  if _n_=1 then do;
    retain w1 w2 w3 w4;
    w1="&week"d;
    w2=intnx("WEEK", w1, -1);
    w3=intnx("WEEK", w1, -2);
    w4=intnx("WEEK", w1, -3);
  end;
  select(week);
```

```
      when(w1) do;
        w1_lat=lat;
        w1_lon=lon;
      end;
      when(w2) do;
        w2_lat=lat;
        w2_lon=lon;
      end;
      when(w3) do;
        w3_lat=lat;
        w3_lon=lon;
      end;
      when(w4) do;
        w4_lat=lat;
        w4_lon=lon;
      end;
      otherwise do;
        if not lim then delete;
      end;
    end;
  run;
```

These sets of lat/lon pairs allow for visualization where the most recently edited nodes are bright yellow, the previous week slightly darker etc. Using a set of SCATTER statements allows the colors to be controlled directly:

```
  proc sgmap plotdata=work.plotdata noautolegend;
    esrimap url="&esri_url/World_Dark_Gray_Base";
    title h=2 "Democratic Republic of the Congo: OSM Node Edits";
    title2 h=1.5 "%sysfunc(propcase(%substr(&week,3)))";
    scatter x=x y=y / transparency=1.0;
    scatter x=w1_lon y=w1_lat / markerattrs=(symbol=circle color=cxFFFF22
                                            size=1px);
    scatter x=w2_lon y=w2_lat / markerattrs=(symbol=circle color=cxC5C517
                                            size=1px);
    scatter x=w3_lon y=w3_lat / markerattrs=(symbol=circle color=cx8A8A0B
                                            size=1px);
    scatter x=w4_lon y=w4_lat / markerattrs=(symbol=circle color=cx505000
                                            size=1px);
  run;
```

The image in Output 13 shows the results of the PROC SGMAP code, with a different ESRIMAP theme of World_Dark_Gray_Base and the brightest areas showing the current week edits:



Output 13. OpenStreetMap node edits by week

The following Imagemagick command will convert the files named 04_drc_hexmonth*.png into 04_drc_hexmonth.mov with a specified delay between frames and importantly a morph factor which specifies the number of intermediate frames for smooth animation:

```
convert -delay 5 -morph 5 04_drc_plotweek*.png -loop 0 miff:- | convert -
04_drc_plotweek.mov
```

The resulting animation can be viewed on YouTube at https://youtu.be/bNx4RK6TPrA. and shows the edits that occur throughout the Democratic Republic of the Congo over the period of analysis.

CONCLUSION

In 1854 John Snow made medical history and changed the way health data is represented by mapping cholera cases across London to visualize the source of the problem. In addition to deaths at residences, he included the source of the problem - contaminated water pumps - providing data points for more effective outbreak control. Since then, maps have been helping aid organizations and medical professionals better respond to emergencies, track the spread of diseases and improve access to healthcare facilities.

Today, collaborative mapping and geospatial data collection is changing the inclusion outcomes for many communities. From disaster risk reduction to the elimination of diseases such as malaria, global volunteers are literally putting communities on the map to change their access to support and services.

From a technical perspective, global contributors to OpenStreetMap result in a rich source of data for analysis that can be used for many purposes. Just-in-time mapping provides on-the-ground volunteers with accurate, detailed maps of hard-to-reach communities. Proactive mapping of infrastructure and risk assessment is assisting with early-warning and disaster management preparation.

SAS Software includes the functionality to extract, analyze and visualize data from OpenStreetMap. The data can be sourced from APIs (using PROC HTTP) or OpenStreetMap downloads, and then converted into SAS dataset format from the XML or JSON data formats. SAS Software includes many tools for analysis and visualization of the data,

including the ability to use OpenStreetMap tile servers as the background for map displays with PROC SGMAP and SAS® Visual Analytics.

The repository https://github.com/m-matthews/hot-sas contains the complete source code and files used to perform the analyses in this paper.

REFERENCES

OpenStreetMap. "Copyright and License." Accessed February 25, 2019. https://www.openstreetmap.org/copyright

OpenStreetMap Wiki. "Planet OSM." Accessed February 25, 2019. https://wiki.openstreetmap.org/wiki/Planet.osm

OpenStreetMap Wiki. "Overpass API." Accessed February 25, 2019. https://wiki.openstreetmap.org/wiki/Overpass_API

OpenStreetMap Wiki. "Osmconvert." Accessed February 25, 2019. https://wiki.openstreetmap.org/wiki/Osmconvert

OpenStreetMap Wiki. "Osmfilter." Accessed February 25, 2019. https://wiki.openstreetmap.org/wiki/Osmfilter

OpenStreetMap Wiki. "Osmosis." Accessed February 25, 2019. https://wiki.openstreetmap.org/wiki/Osmosis

Humanitarian OpenStreetMapTeam. "What We Do." Accessed February 25, 2019. https://www.hotosm.org/what-we-do

Geofabrik. "Downloads." Accessed February 25, 2019. https://www.geofabrik.de/data/download.html

CONTACT INFORMATION

Your comments and questions are valued and encouraged. Contact the author at:

Michael Matthews
michaelm@innotwist.com
https://github.com/m-matthews/hot-sas