LearnOSM

Reviewing OSM Data

Reviewed 2017-04-24

This section covers processes for checking data quality, particularly in the context of a directed OSM mapping project, such as those undertaken by the Humanitarian OpenStreetMap Team in various countries and Open Cities projects in Bangladesh, Sri Lanka, and Nepal. The methods demonstrated may be useful in other contexts as well, when data quality review is a regular task.

When we are trying to map a complete set of features and attributes in a specified area, we need ways to check for mistakes and ways to assess the accuracy of the work. In this tutorial we will work through several methods of checking data, explaining the steps of the method and the reason behind each. A well-managed mapping project will include each of these three processes, both for evaluating and correcting data and for reporting.

- Daily Checks
- Re-Surveying
- SQL Queries

These methods of review become more important as the data model grows and the number of features collected becomes quite large. For example, it would not take a lot of time and effort to assess a data model which only involves points of interest (POIs):

Data Model	
POIs	Name
	Type
	Phone Number

In this case the questions to ask would be:

- Have the POIs been mapped in all locations?
- Are any POIs missing the name attribute?
- Are any POIs missing the type attribute?
- Are any POIs missing the phone number attribute?
- Is the value in the name field correctly capitalized?
- Does the phone number make sense?

Usually a data model is far more complex, however, as in the case with mapping buildings. Consider a data model that includes this:

Buildings	Name
	Street Name
	Building Number
	Number of Levels
	Use of Building
	Material of Building
	Type of Structure
	Use of Building
	Construction Date
	Building Condition

Now you may be mapping thousands of buildings that have many attributes, and the analysis becomes more critical. In this tutorial we will use buildings as an example, though the same methods can be applied for reviewing other types of features too.

Daily Checks

The most immediate way to check data is to review and validate it on a regular basis. This could be daily or at most weekly. For the supervisor of a team of mappers, this is an important task because catching mistakes and bad editing practices early means that they can be corrected and the editors can learn to do things properly.

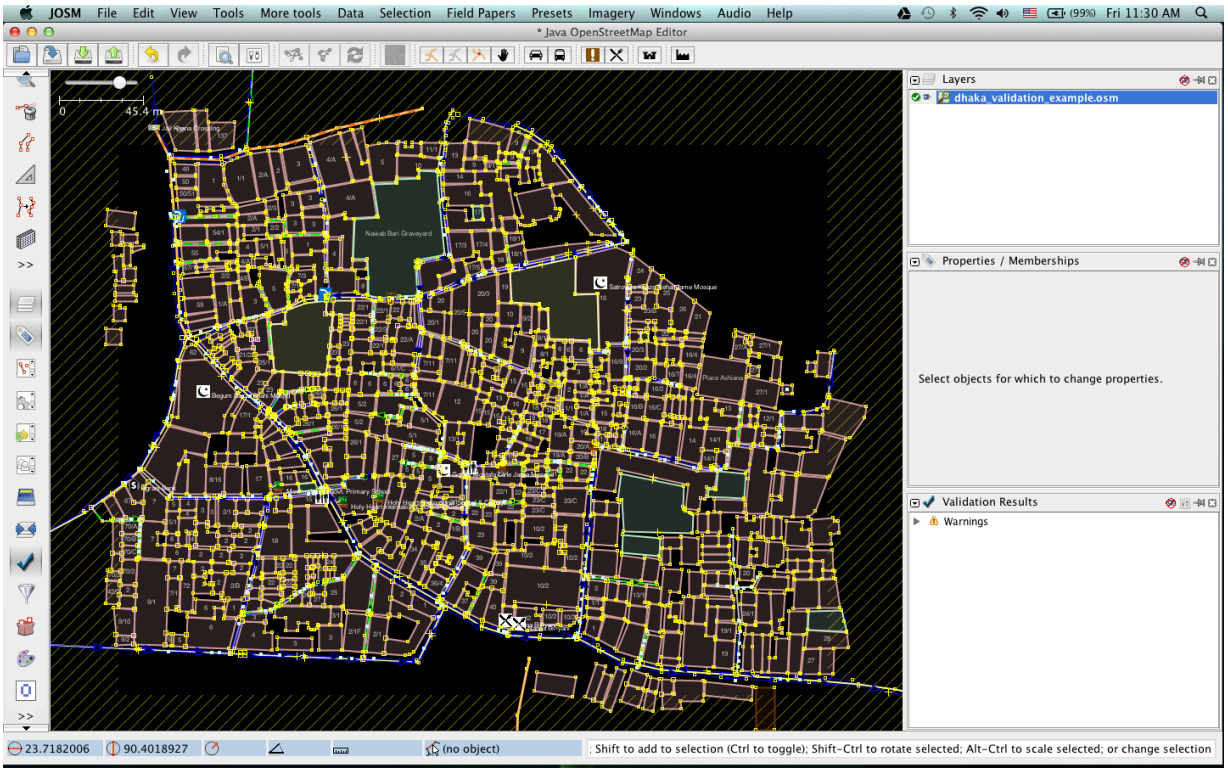
Here we will look at some methods for checking data simply using JOSM. Some of the questions we are asking about our data are:

- Are there ***topology*** errors (like overlapping buildings or incorrect relations)?
- Are there ***tagging*** errors (misspelled tags, misused key-value combinations)?
- Is the data ***complete*** according to the data model?

Let's examine how we can find answers to these questions in JOSM. We'll assume that we are examining the work of others, but the same processes will work fine (and should be easier) when analyzing your own work.

We will be using an example data file from the Open Cities mapping project in Dhaka. To follow along, download the following file: dhaka_validation_example.osm

DO NOT try to save your changes on OpenStreetMap. These exercises are for demonstration purposes only.

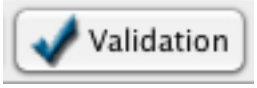


Data Validation

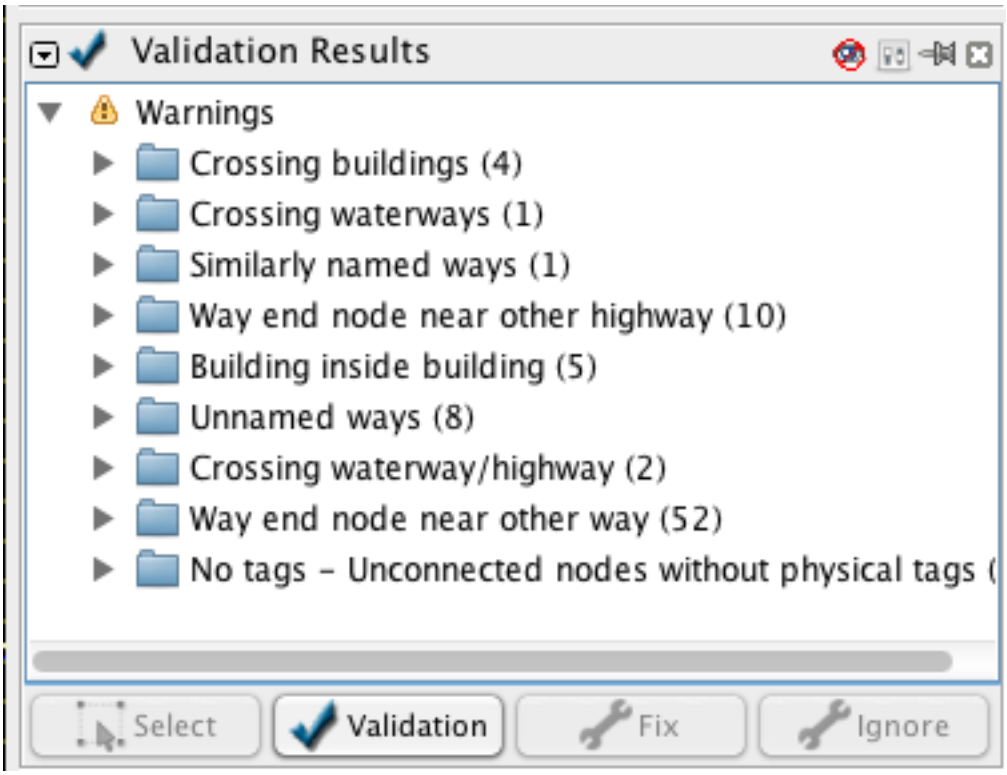
The first step for checking data is to run the Validation tool in JOSM, which will automatically check the data you have open for suspected mistakes. This tool is especially useful for finding ***topology*** errors but may not be as useful for finding incorrect tags.

- Activate the tool by clicking on the Validation Tool button on the left side of JOSM. (This is unnecessary if the Validation panel is already open)

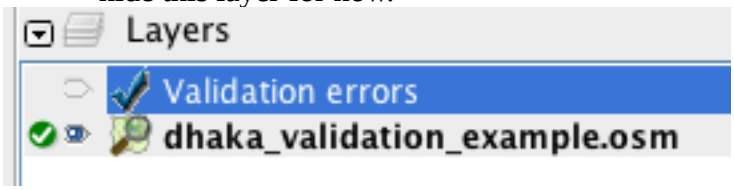
- Next make sure that nothing is selected by clicking in a blank spot on your map. If you have features selected when you run the Validation Tool, only those selected features will be checked. (sometimes you may want to only check certain features, but for now we will check the entire file)
- Click the “Validation” button on the panel.



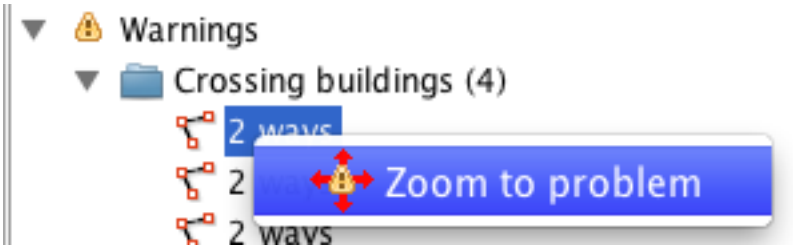
- You will see a list of warnings appear:



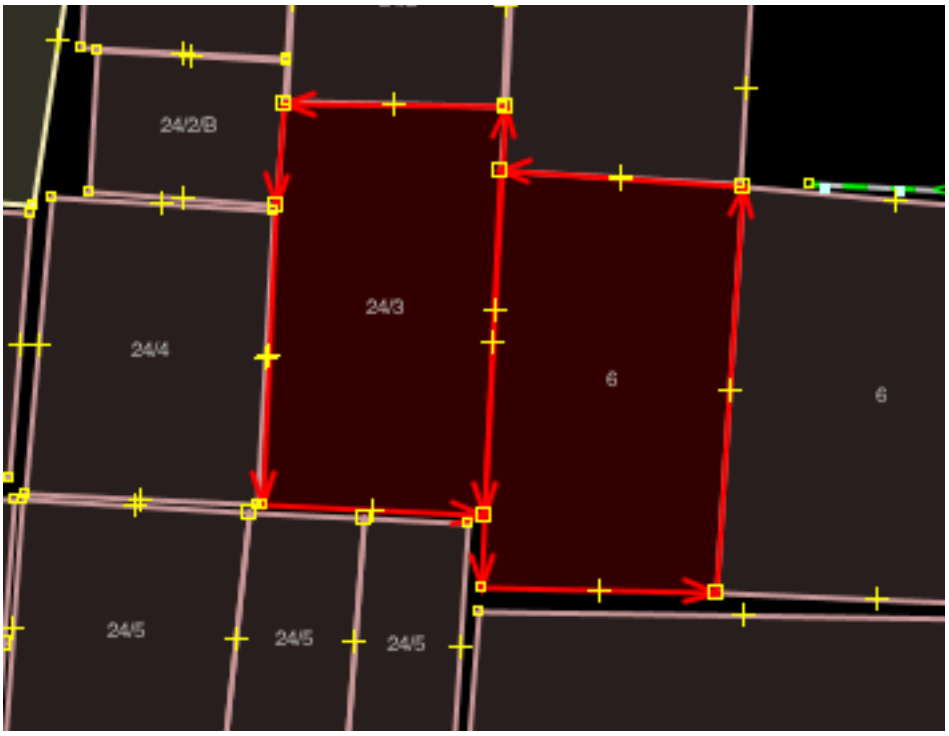
- A new layer also appears, showing where the errors are. You may find it convenient to hide this layer for now.



Let's look at a few of the warnings. You can see that there are four "Crossing buildings" warnings. This warning means that buildings are overlapping somewhere. Select the first item in this list, right-click, and click "Zoom to problem."



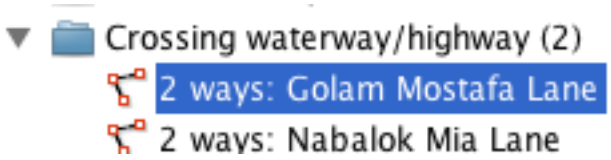
Also, click the "Select" button at the bottom of the Validation window to select the ways in question. This shows that these two ways have a problem:



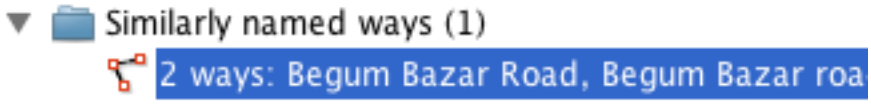
- This is an error that we never would have caught without the validation tool. If you zoom in very close you can see that there is a tiny overlap between the buildings, which is a topological error, because buildings don't usually overlap with each other. To fix this, the middle node needs to be moved. If the buildings are actually touching, which they probably are, the middle node can be joined to the way.
- Once this is corrected, we can run the Validation tool again and this warning will have disappeared from the list.

This method of automatically checking the data is an effective way of correcting topology errors, particularly those that would be difficult to notice by a person. In the list of validation warnings, you can see that other warnings such as "Building inside building" is the result of a similar mistake.

Still other warnings, such as "Crossing waterway/highway," are not necessarily mistakes. This shows that the validation tool is good at finding possible mistakes, but it requires someone to go and see whether the error is important or not.



Let's look at the warning under "Similarly named ways" to see an error that is not topological. Click "Select" to select the two ways in question.

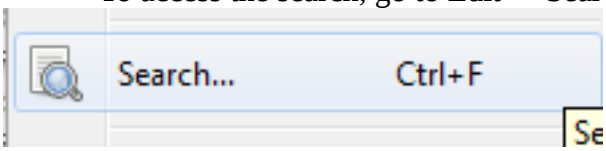


Can you tell what the mistake is? Here we have two different road segments, which are actually the same road, yet they have been named slightly differently - “road” is capitalized on one of the ways but not on the other. It makes sense that they should have the same name, and in this case the word “road” should be capitalized.

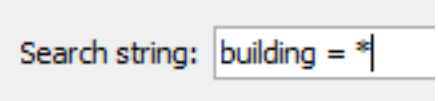
Using JOSM Search

Searching in JOSM is a powerful way of reviewing data. It allows you to provide search terms, also known as queries, to select only the features that you want.

- To access the search, go to Edit -> Search or press CTRL + F on your keyboard.



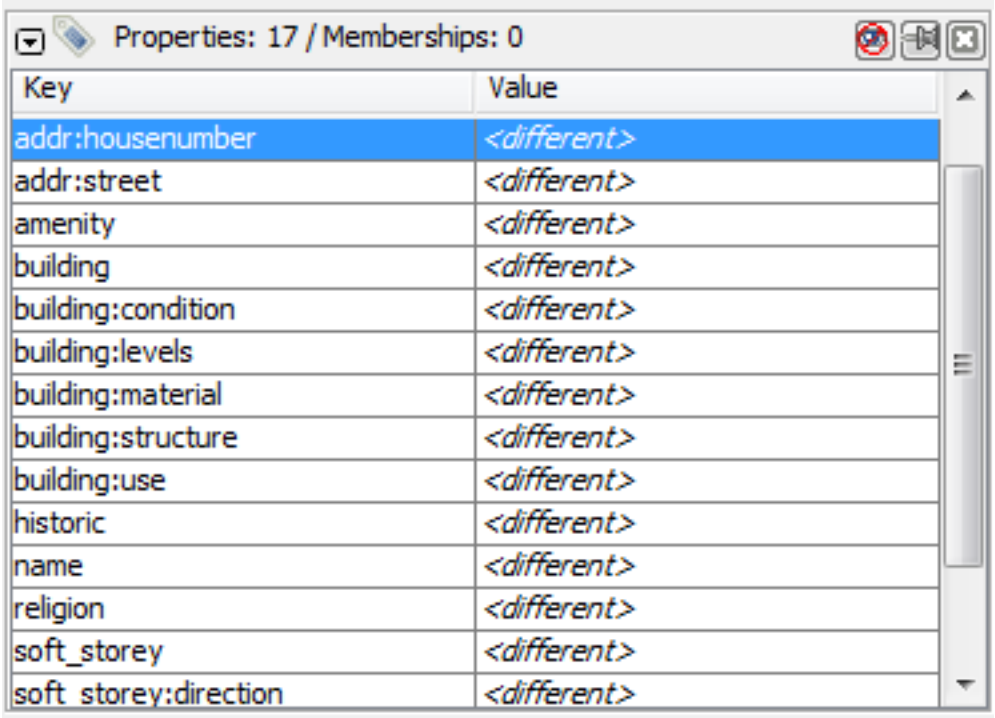
- There are a great many types of query you can search here, and you can see details and examples in the search box itself and by clicking on the “Help” button.
- For now, let’s try to select all of the buildings. Almost every building is going to have the tag ***building=yes*** and a few will have ***building=construction***. This means that we can build a query that reads:
  building = yes OR *building=construction*
- This should select all the buildings, but just in case somebody applied the wrong tag to a building, we can use a wildcard character instead, which will select all features that have the key ***building***.



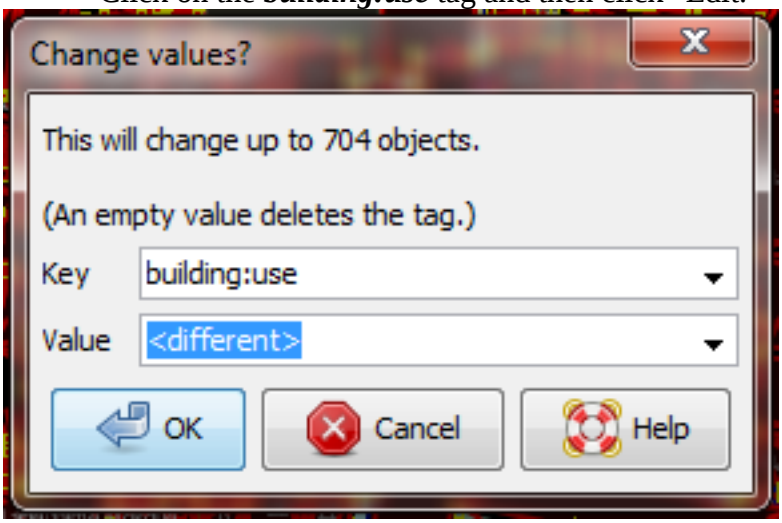
- All of the buildings will be selected.

This is great, but how does it help us review the data? Well, now that all of a single type of feature have been selected, we can look for incorrect tags.

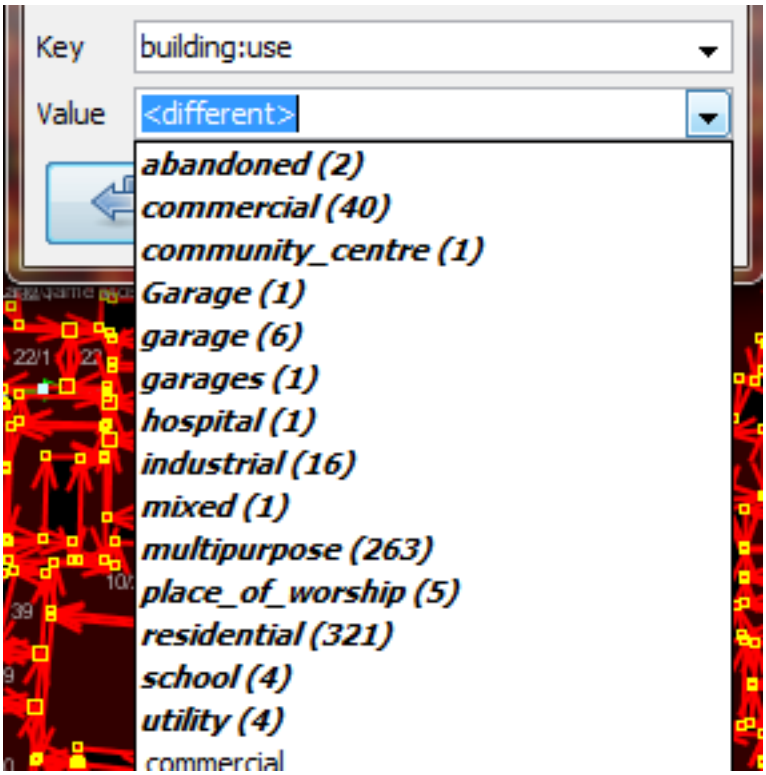
- Look in the Properties window - what we see is all of the tags for every selected object. They all share the same keys, but because each feature has different values they are marked as *<different>*.



- Click on the ***building:use*** tag and then click "Edit."



- BE CAREFUL HERE! You don't want to edit the value and click OK, because that will change this tag for every single building feature. **This would be very bad.**
- Instead, click on the drop-down box next to Value.



- Notice that all of the items in bold have a number next to them in parentheses. This is the number of selected features that have this tag value.

We can compare this with the OpenStreetMap tags that have been mapped in our data model, and look for mistakes. For example, this tag represents the use of the building. Early in the Open Cities Dhaka project (where this data came from) there was uncertainty as to whether a mixed-use building should be tagged ***building:use=multipurpose*** or ***building:use=mixed***. Because the former tag had been used previously in other countries, it was selected. However, we see here that one of the buildings has been tagged as ***mixed***. We need to correct this. (Another obvious mistake are the three different terms for ***garage***, but we won't correct this here.)

- We cannot change the feature that has the ***building:use=mixed*** tag here, because we have hundreds of features selected. So, to correct the mistake, we must first find it. How? You guessed it - with the Search tool.
- Click "Cancel" to exit this dialog. **Remember, clicking OK can be dangerous.**
- Open the Search again and enter the following query: *"building:use"=mixed*
- Note that the quotation marks are necessary because a colon (:) has a search meaning as well. This should cause the one building that has this tag to be selected. It can now be corrected to the value ***multipurpose***.

Remember that if you are following along with this tutorial, DO NOT try to save your changes on OpenStreetMap. These exercises are for demonstration purposes only.

Re-Surveying

When managing a project like a detailed building survey, there ought to be an additional method of quality control, both for improving the work and for reporting on the accuracy at the end of a

project.

If there are many mapping teams collaborating to survey an area, it is common that one or more of the teams may not do a satisfactory job. Even those teams that do efficient and accurate work will make mistakes. Imagine teams that each map 100 buildings per day - it is not unlikely that a small percentage of the attributes they collect may be incorrect.

Thus, a good project will include a process of re-checking some of the work that has been done, fixing mistakes, determining which mapping teams are performing satisfactorily, and approximating the percentage of errors for a a final report.

Of course, there is no sense in re-surveying every building in a target area, but 5-10% of the buildings should be reviewed. The areas for review should be chosen from different areas to compare between survey teams. Survey teams can re-survey each others' work, or if possible more experienced managers can undertake the reviews. It is common practice that one day a week managers will spend re-surveying parts of the target area.

Correcting Mistakes

What should be done when mistakes are found?

If there is a small amount of mistakes (less than 5% of buildings), the issues should be brought to the original mapping team so that they are aware and may not make the same mistakes again. The data should be corrected in OpenStreetMap and the results of the re-survey should be recorded.

If there are many mistakes, bigger actions may need to be taken. The survey team will need to be addressed in an appropriate fashion, and the areas they have mapped may even need to be resurveyed entirely, depending on how inaccurate the data proves to be. Greater than 10% inaccuracy is most likely an unacceptable rate.

Reporting on Accuracy

The second goal of resurveying is so that you can report on the accuracy of the data when the project closes. Users of the data will want to know your metrics and methodologies of assessing the data quality.

By including this process as part of your reviewing methodology, you will be able to clearly explain how you assessed the data quality, and provide hard numbers that show the likely percentage of error contained in your survey data.

For example, let's imagine that we are managing a project which maps 1000 buildings. So we decide to map 10% of them, or 100 buildings, randomly selected from the target area. We go out and find that of the 100 buildings we resurveyed, six of them have a high level of inaccuracy. Let's say we define inaccuracy by having more than one attribute incorrect. So six percent of the resurvey is wrong - we can fix these mistakes, but we still must extrapolate that about six percent of all 1000 buildings are probably inaccurate. This should be reported as the probable error at the close of the project.

Resurveying ought to be done throughout the project. Imagine that we waited until the end in this example and 40 out of 100 buildings were wrong! It might ruin the entire project. It is better to catch large-scale mistakes early so that they can be corrected.

SQL Queries

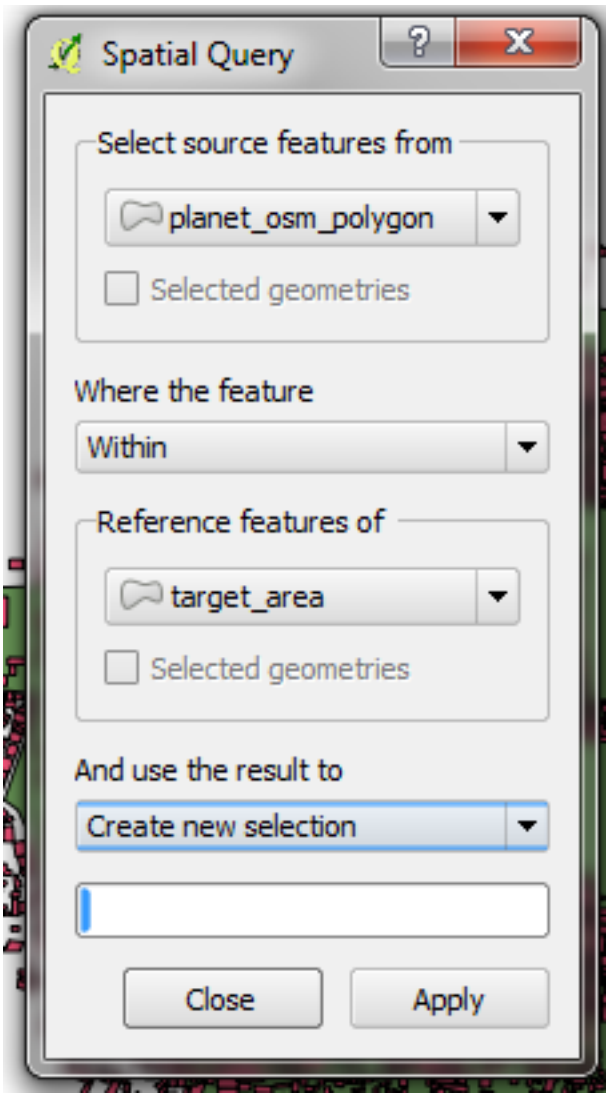
Probably the best analysis tool is going to be running SQL Queries in a GIS system, such as Quantum GIS. This is similar to searching for data in JOSM, but it offers more powerful analysis, though it can take a little more time to set up. Using JOSM is a quick, regular way to check for basic errors, whereas querying in QGIS is better suited for finding missing data or incorrect attributes.

We'll assume here that you are somewhat familiar with GIS, and focus on building queries which can help you to review OpenStreetMap data. For the exercises below we'll again be using data from the Open Cities Dhaka project, which you can download at dhaka_sql.zip. The OpenStreetMap data was exported using the HOT Export Tool (export.hotosm.org) and the target area boundary was defined at the start of the project.

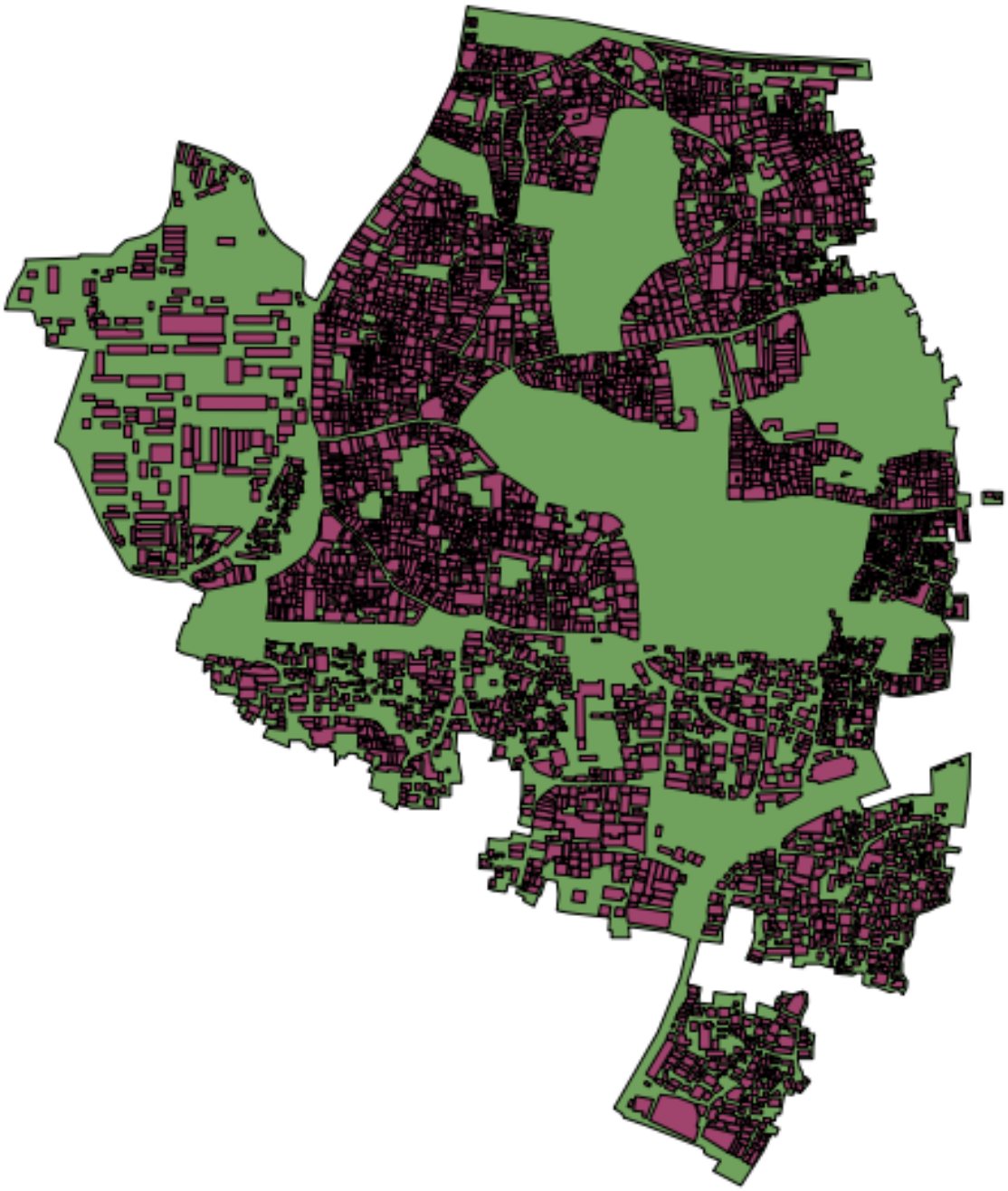
Prepare the Data

Unzip the files and load the two shapefiles into QGIS. We'll begin by clipping only the buildings within the project area, to make our queries more simple later on.

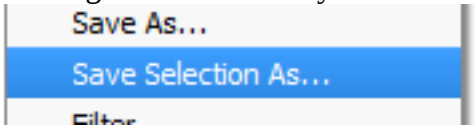
- First let's select only the polygons that are within the target area. To do this we will use the Spatial Query Plugin. If you do not already have it installed, go to Plugins -> Manage and Install Plugins to find and install it.
- Go to Vector -> Spatial Query -> Spatial Query.
- You should fill in settings to select features from **planet_osm_polygon** that are **within target_area**.



- Click Apply. Only polygons within the target area will be selected.



- Right-click on the layer and save the selection as a new shapefile. Add it to the project.

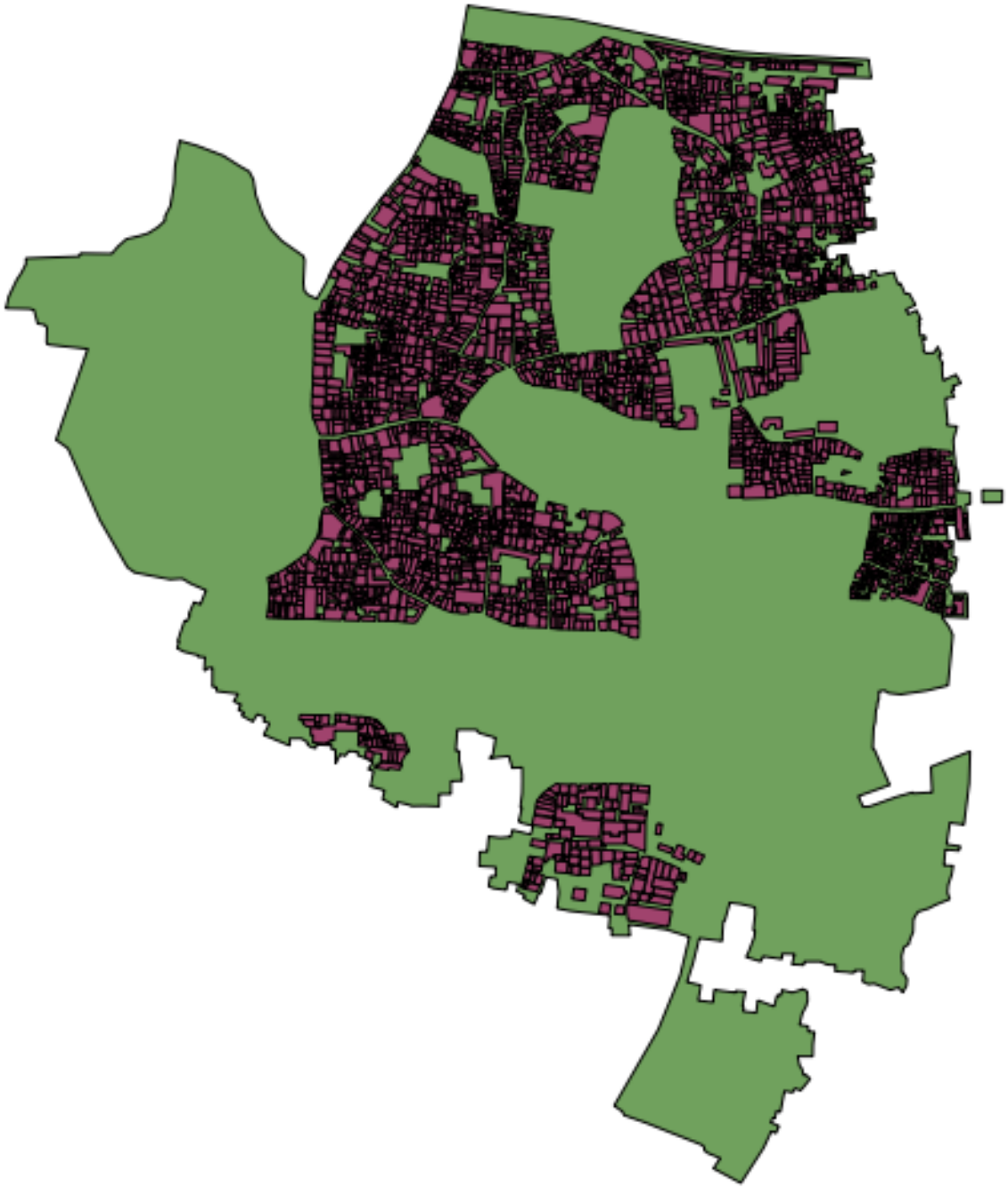


- Next let's filter only the polygons that are buildings and that were collected as part of this project.
- Right-click on **planet_osm_polygon** and click on "Filter…"

- Enter the following query:

“building” != NULL AND “source” = ‘Open Cities Dhaka Survey’

- Click OK. Filtering the data with this query will only show polygons that have something in the building column. It also removes buildings that do not have the source tag associated with this project.
- Save this data as a new shapefile. We will use this file for our SQL queries.



SQL Queries

We can now run queries on the buildings layer to find possible mistakes. Let's think about some things that we might want to query. The data model from this project indicates attributes that should be collected for every building - they are:

- name
- building
- building:levels
- building:use
- building:vertical_irregularity
- building:soft_storey
- building:material
- building:structure
- start_date
- building:condition

Note that in the shapefile these attribute names are truncated, since column named are limited to 10 characters.

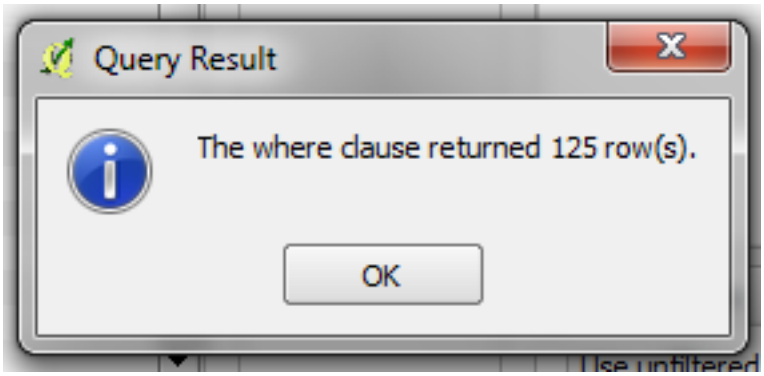
So what sort of questions do we want to ask? What are likely mistakes? One common mistake is that a building was mapped, but not all of the attributes were collected. So we will want to run a query that shows all the buildings which do not have a complete set of attributes. Of course, for some attributes, like name and start_date (construction year), it is perfectly fine for them to be empty, because not every building has a name and sometimes the construction year is unknown. But the other attributes should always be collected.

Let's try to develop a query for this:

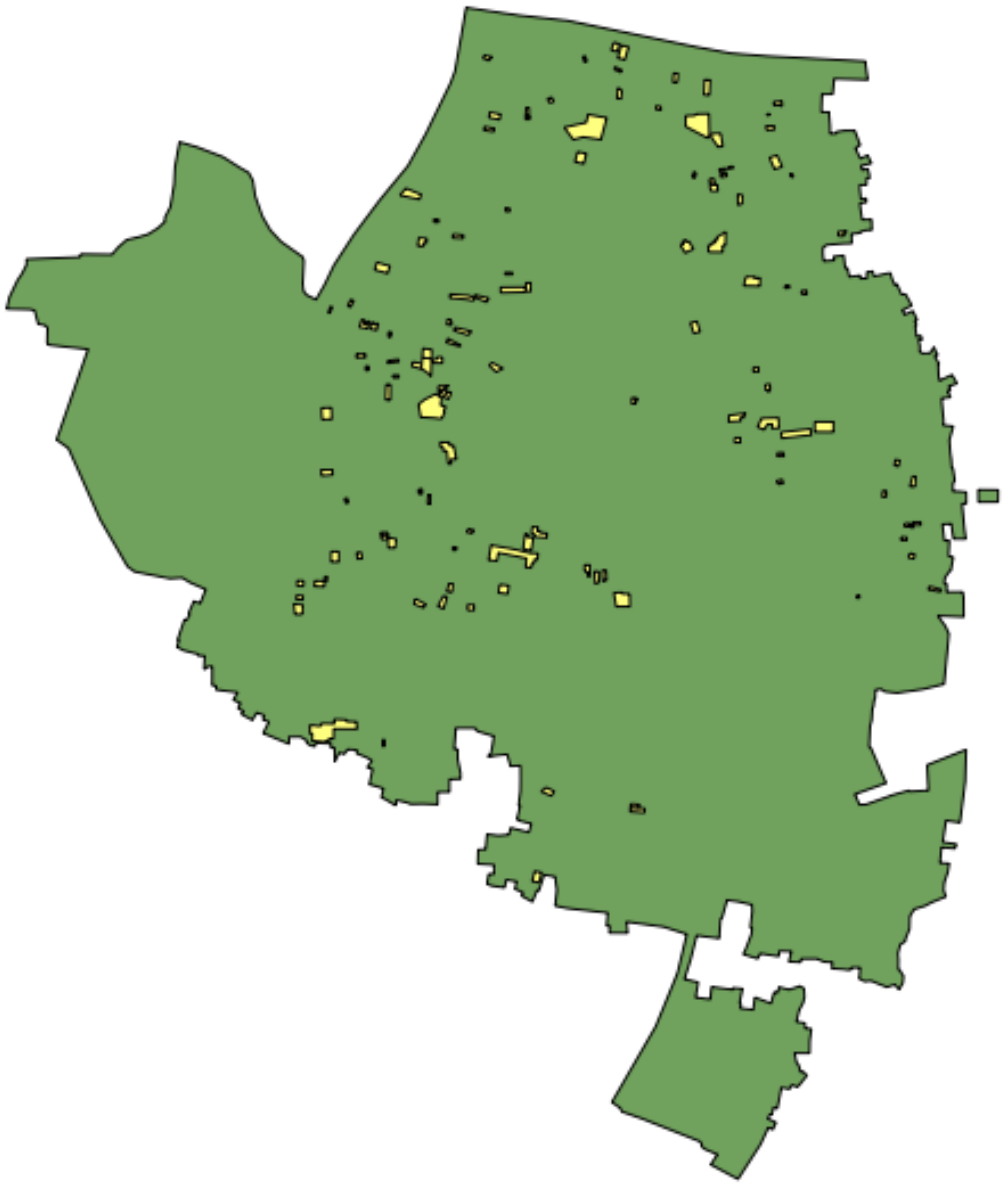
- Right-click on the buildings layer (the layer we created in the previous section) and click "Filter…" This will open the query builder. Here we can write complex queries to return only the data we want.
- You can build your own query by double-clicking on the fields, operators, and values, or you can copy the query that we have built here:

"building_c" = NULL OR "building_s" = NULL OR "building_l" = NULL OR "building_m" = NULL OR "vertical_i" = NULL OR "soft_store" = NULL OR "building_u" = NULL

- Click "Test" and you will see that this query returns 125 features. This means that of the 3500 or so buildings that have been mapped, 125 of them are missing one or more of these attributes.



- Click OK to show only those buildings that meet the conditions in the query.



- These buildings can then be examined more closely to identify which attributes are missing, and if they need to be resurveyed. It is then possible to use QGIS to create a map which a re-survey team could take to correct the missing building attributes.

What are some other queries that might be of use? Well, you may also want to check for attributes that are not contained within your data schema. We did this in the JOSM search section. You can use a query to find all the buildings whose attributes don't fit within your data model.

You may also use this to look for anomalies, which are probably but not necessarily mistakes. For example, if we open the query builder, select **building_l**, and click "All" to load all the possible attribute values, we see that most buildings have a number between one and 20 (This attribute is building:levels, the number of storeys in the building). But there is also a 51 in there. It seems unlikely that there will be a 51 storey building towering above everything in this area, so we can locate it and make a note to check this with the mappers.

Querying can be an effective way to look for possible mistakes in the data set. Combined with other features of QGIS, it can be used to output maps that can be used for reviewing the data in an area.

Summary

In this tutorial we've gone through several effective methods of maintaining data quality during a project and done some hands-on exercises to practice reviewing OSM data. When organizing a mapping project, or even when assessing the data in an area for personal use, these methods may come in handy.

Was this chapter helpful? Let us know and help us improve the guides!

- learnosm@hotosm.org
- @learnOSM
- Hosted on Github

Humanitarian OpenStreetMap Team