



OPENQUAKE
calculate share explore

| Parroquia | QDNOELECTR | QDNOWATER | QNOBATH | HOUDENSITY | QMANUINDUS | QCOMMERC |
|---------------|------------|-----------|-----------|------------|------------|----------|
| GUARANDA | 33.613313 | 49.083941 | 16.673735 | 36.669129 | 3.535945 | 10.58180 |
| ACUNDO V... | 54.763537 | 79.712132 | 45.235362 | 8.996732 | 1.579670 | 0.96153 |
| JULIO E. M... | 59.244992 | 81.510015 | 53.394256 | 15.463391 | 0.594307 | 0.83472 |
| ALINAS | 54.011345 | 75.526742 | 30.958904 | 5.255120 | 7.889348 | 3.37328 |
| W LOREN... | 38.790932 | 71.536524 | 26.873857 | 8.365859 | 0.375000 | 1.75000 |
| N SIMON... | 37.206179 | 62.323707 | 20.498094 | 15.404406 | | |

INTEGRATED RISK MODELLING TOOLKIT
USER INSTRUCTION MANUAL

Hands-on-instructions on the different functionalities of the Integrated Risk Modelling Toolkit



User Manual

Release 3.25.1

GEM Foundation

Mar 02, 2026

CONTENTS

| | | |
|-----------|---|-----------|
| 1 | Installation and troubleshooting | 2 |
| 1.1 | How to run tests and build the documentation | 3 |
| 2 | Introduction | 4 |
| 3 | Definitions and acronyms | 5 |
| 4 | The plugin menu | 6 |
| 5 | OpenQuake IRMT settings | 7 |
| 5.1 | Advanced settings | 8 |
| 6 | Driving the OpenQuake Engine | 9 |
| 6.1 | Run a calculation | 10 |
| 6.2 | The list of calculations | 10 |
| 6.3 | Watch the console log | 10 |
| 6.4 | Download outputs | 10 |
| 6.5 | Run a postprocessing (or risk) calculation on top of a previously computed hazard . . . | 12 |
| 6.6 | Remove a calculation | 12 |
| 6.7 | Abort a calculation | 12 |
| 6.8 | Calculation status | 13 |
| 7 | The OpenQuake IRMT Data Viewer | 14 |
| 7.1 | Visualizing outputs of hazard calculations | 14 |
| 7.2 | Visualizing outputs of risk calculations | 18 |
| 8 | Visualizing OpenQuake Volcano Scenario Calculator outputs | 19 |
| 8.1 | Visualizing Exposure + Risk | 20 |
| 8.2 | Visualizing Input Files | 22 |
| 8.3 | Visualizing Total Risk | 23 |
| 9 | Aggregating points by zone | 24 |
| 10 | Transforming attributes | 25 |
| 11 | Leveraging R to perform statistical analysis in QGIS | 28 |
| | Bibliography | 33 |

Contents:

INSTALLATION AND TROUBLESHOOTING

This plugin runs with [QGIS 3.0¹](#) and above.

On Microsoft Windows, QGIS includes all the software dependencies needed by the plugin.

On macOS and Linux, please make sure that [Matplotlib²](#) with Qt5 backend is installed.

On macOS:

```
$ python3 -m pip install --upgrade pip
$ python3 -m pip install --upgrade matplotlib
```

On Ubuntu (Debian and similar):

```
$ sudo apt install python3-matplotlib python3-pyqt5.qtwebkit
```

On Fedora and similar:

```
$ sudo dnf install python3-matplotlib python3-matplotlib-qt5
```

On macOS make sure to run the script located under */Applications/Python X.Y/Install Certificates.command*, after Python has been installed, to update the SSL certificates bundle.

On Fedora, please follow these [instructions³](#).

The plugin can be installed using the QGIS Plugins Manager, that is accessible through the QGIS menu as *Plugins -> Manage and install plugins*. Please note that the *Settings* of the Plugins Manager contain a checkbox to *Show also experimental plugins*. If that option is checked, the latest version of the plugin that is marked as *experimental* will be available for installation. Otherwise, the latest *stable* version will be installable.

Note: The latest *stable* version of the plugin is compatible with the Long Term Support version of the OpenQuake engine, whereas the latest *experimental* version of the plugin is aligned with the latest version of the OpenQuake engine.

Experimental versions contain new functionalities that may have not been properly tested yet, and that could cause the plugin or QGIS to break or to behave unexpectedly.

Some users reported issues about *upgrading* the plugin to its latest version. We recommend to *reinstall* the plugin instead, in order to make sure the new installation is done in a clean folder.

¹ <http://qgis.org/it/site/forusers/alldownloads.html>

² <https://matplotlib.org/users/installing.html>

³ <https://copr.fedorainfracloud.org/coprs/dani/qgis/>

1.1 How to run tests and build the documentation

Tests run on a QGIS docker container based on Ubuntu Xenial (16.04LTS), launching one of the following commands from the *svir* directory:

```
$ ../scripts/run_unit_tests.sh  
$ ../scripts/run_integration_tests.sh
```

The user manual, both in the html and pdf versions can be built by the same QGIS docker container running the following command from the *svir* directory:

```
$ ../scripts/make_doc.sh
```

INTRODUCTION

At the core of the Global Earthquake Model (GEM) is the development of state-of-the-art modeling capabilities and a suite of software tools that can be utilized worldwide for the assessment and communication of earthquake risk. For a more holistic assessment of the scale and consequences of earthquake impacts, a set of methods, metrics, and tools are incorporated into the GEM modelling framework to assess earthquake impact potential beyond direct physical impacts and loss of life. This is because with increased exposure of people, livelihoods, and property to earthquakes, the potential for social and economic impacts of earthquakes cannot be ignored. Not only is it vital to evaluate and benchmark the conditions within social systems that lead to adverse earthquake impacts and loss, it is equally important to measure the capacity of populations to respond to damaging events and to provide a set of metrics for priority setting and decision-making.

The employment of a methodology and workflow necessary for the evaluation of seismic risk that is integrated and holistic begins with the OpenQuake Integrated Risk Modelling Toolkit (OpenQuake IRMT). The OpenQuake IRMT is QGIS plugin that was developed by the [Global Earthquake Model \(GEM\) Foundation](http://www.globalquakemodel.org/)⁴ and co-designed by GEM and the [Center for Disaster Management and Risk Reduction Technology \(CEDIM\)](https://www.cedim.de/english/index.php)⁵.

The OpenQuake IRMT plugin has evolved significantly with respect to its original purposes, in order to make it operate seamlessly with the [OpenQuake Engine](https://github.com/gem/oq-engine)⁶ (OQ-engine) ([PMW+14] and [SCP+14]). This enables a whole end-to-end workflow, where calculations of physical hazard and risk can be run directly from within the QGIS environment (see [Driving the OpenQuake Engine](#)) and the outputs of such calculations can be loaded as QGIS vector layers. Those of them that can be visualized as maps (e.g. hazard maps) are also automatically styled with respect to fields selected by the user. Others can be plotted as curves (e.g. hazard curves) inside a *Data Viewer* window (see [The OpenQuake IRMT Data Viewer](#)) that was conceived for this purpose.

⁴ <http://www.globalquakemodel.org/>

⁵ <https://www.cedim.de/english/index.php>

⁶ <https://github.com/gem/oq-engine>

DEFINITIONS AND ACRONYMS

In this manual, the terminology *layer*, and *project* are used ubiquitously, and it is important to explain what the terminology means as well as its use. In QGIS, a *project* or *project file* is a kind of container that acts like a folder storing information on file locations of layers and how these layers are displayed in a map. It is the main QGIS datafile. A *layer* is the mechanism used to display geographic datasets in the QGIS software, and layers provide the data that is manipulated within the OpenQuake IRMT. Each layer references a specific dataset and specifies how that dataset is portrayed within the map. The standard layer format for the OpenQuake IRMT is the ESRI Shapefile [ESRI98] that can be imported within the QGIS software using the default *add data* functionality, or layers may be created on-the-fly within the OpenQuake IRMT using GEM socio-economic databases. A QGIS project can include multiple layers that can be utilized to provide the variables and maps necessary for an integrated risk assessment.

In seismic hazard and risk analysis, a Ground-Motion Measure Type (GMMT) is a physical quantity expressing a particular characteristic of the ground shaking recorded or computed at one site. The most important and frequently used GMMTs are scalar and they indicate the shaking intensity (i.e. the amplitude of the ground shaking). These GMMT are also indicated with the acronym IMT. Other ground-motion measures define, for example, the duration of the shaking. A very common IMT is the peak ground acceleration, specified with the PGA acronym. The principal IMT used in the OpenQuake-engine are:

- Peak Ground Acceleration – indicated as PGA – measured in fractions of g
- Peak Ground Velocity – indicated as PGV – measured in cm/s
- Peak Ground Displacement – indicated as PGV – measured in cm
- Spectral Acceleration for a given period T – indicated as Sa(T) - measured in fractions of g

An Intensity Measure Level (IML) is a value of a specific IMT.

THE PLUGIN MENU

When the OpenQuake IRMT plugin is installed, it adds its own menu to those available on the QGIS graphical user interface. The plugin menu contains the options listed below. For each option, this manual provides a separate chapter with the description of its functionality and of the typical workflows in which it is used. Please follow the links next to the option icons, to reach the corresponding documentation.

Note: The menu options are disabled when the corresponding functionalities can not be performed. For instance, the *Transform attributes* option will be available only as long as one of the loaded layers is activated.

- OpenQuake Engine

-  *Driving the OpenQuake Engine*


- Utilities

-  *Aggregating points by zone*

-  *Transforming attributes*

-  *OpenQuake IRMT settings*

-  *The OpenQuake IRMT Data Viewer*

-  OpenQuake IRMT manual: a web browser will be opened, showing the html version of this manual

OPENQUAKE IRMT SETTINGS

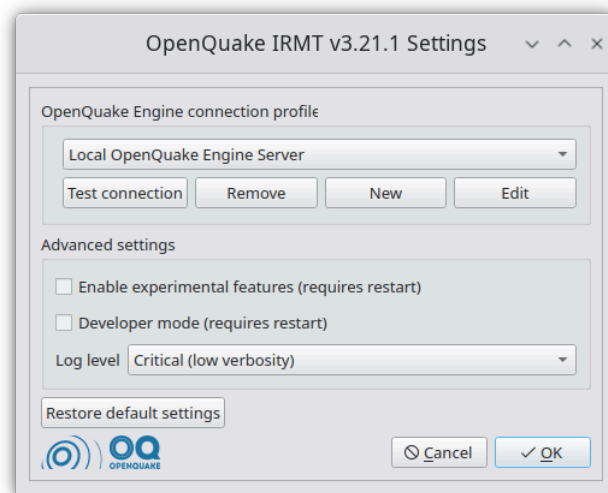


Fig. 5.1:  OpenQuake IRMT settings

Some of the functionalities provided by the plugin, such as the ability to work with GEM data, require the interaction between the plugin itself and the OpenQuake Engine (OQ-Engine).

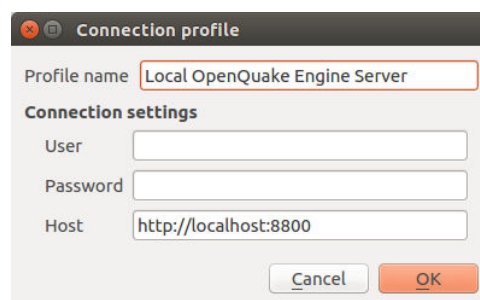


Fig. 5.2: Connection profile

The **OQ-Engine**⁷ is an open source application that allows users to compute seismic hazard and seismic risk of earthquakes on a global scale. It can run locally on the user's machine, or on a remote workstation or cluster. The **OQ Engine Server**⁸ provides an HTTP RESTful API. The OpenQuake IRMT plugin leverages this API to enable the user to drive the OQ-Engine directly from within QGIS. In order to

⁷ <https://github.com/gem/oq-engine>

⁸ <https://github.com/gem/oq-engine/blob/master/doc/running/server.md>

interface the plugin with a running OQ Engine Server (WebUI), it is necessary to insert in this dialog the user credentials (if they are required by the server) and the web URL where the service is hosted (see [Fig. 5.2](#)). By default, the host is set to `http://localhost:8800`, meaning that the plugin will attempt to connect to a OQ Engine Server (WebUI) running locally, on port 8800.

It is possible to create multiple connection profiles, and to edit or remove existing ones. For instance, you might want to use alternatively an OpenQuake Engine Server running on your local machine and a remote one running on a cluster. For this purpose, you can create the two connection profiles, and then select one or the other from the *OpenQuake Engine connection profile* dropdown menu. The buttons *Test connection* can be used to check if the plugin is able to connect to the corresponding server using the specified host name and credentials.

5.1 Advanced settings

The checkbox labeled *Enable experimental features (requires restart)* enables some functionalities of the plugin that are still experimental and not completely tested. It is not recommended for standard users.

It is possible to change the verbosity of the plugin log, by selecting a *Log level* in the corresponding dropdown menu. Levels are:

- Info (high verbosity): whatever is notified to the user through the message box in the graphical user interface, will be also printed in the log.
- Warning (medium verbosity): only critical messages (errors) and warnings will be printed in the log.
- Critical (low verbosity): only critical messages (errors) will be printed in the log.

The checkbox labeled *Developer mode (requires restart)* can be used to log additional information for debugging purposes. It is useful for developers or advanced users for troubleshooting, but it is not recommended for standard users.

The button *Restore default settings* can be used to discard the settings customized by the user and to replace them with the factory defaults. Please be aware that such operation, once confirmed, can not be undone.

DRIVING THE OPENQUAKE ENGINE

Run Calculation


List of calculations

| Description | Job ID | Calculation Mode | Owner | Status | Console | Remove | Outputs | Continue |
|--|--------|------------------|--------|----------|---------|--------|---------|----------|
| Classical PSHA with Characteristic Fault Source defined as sequence of planar fault segments | 12 | classical | travis | complete | Console | Remove | Outputs | Continue |
| Classical PSHA — Area Source | 11 | classical | travis | complete | Console | Remove | Outputs | Continue |
| Scenario Damage and Consequences Demo (Nepal) | 10 | scenario_damage | travis | complete | Console | Remove | Outputs | Continue |
| Scenario Hazard Demo (Nepal) | 9 | scenario | travis | complete | Console | Remove | Outputs | Continue |
| Stochastic Event-Based Risk Demo (Nepal) | 8 | event_based_risk | travis | complete | Console | Remove | Outputs | Continue |
| Stochastic Event-Based Hazard Demo (Nepal) | 7 | event_based | travis | complete | Console | Remove | Outputs | Continue |
| Classical Probabilistic Risk Demo (Nepal) | 6 | classical_risk | travis | complete | Console | Remove | Outputs | Continue |

List of outputs for calculation 10

| Id | Name | Download csv | Download npz | Load layer | Aggregate |
|----|-----------------------|--------------|--------------|------------|-----------|
| 41 | Average Asset Damages | Download csv | Download npz | Load layer | Aggregate |
| 42 | Full Report | Download rst | Show | | |
| 43 | Average Asset Losses | Download csv | Download npz | Load layer | Aggregate |
| 44 | Realizations | Download csv | | | |

Download HDF5 datastore for calculation 10 Show parameters for calculation 10

Fig. 6.1:  Dialog to run the OpenQuake Engine server

The plugin enables to drive the [OpenQuake Engine](https://github.com/gem/oq-engine)⁹, submitting new jobs, watching calculation progress, retrieving and visualizing results, seamlessly within the QGIS interface. This is made possible by leveraging the OpenQuake Engine Server [HTTP RESTful API](https://github.com/gem/oq-engine/blob/master/doc/web-api.md)¹⁰. The connection with a running [OQ-Engine Server \(WebUI\)](https://github.com/gem/oq-engine/blob/master/doc/running/server.md)¹¹ has to be properly set up as described in *OpenQuake IRMT settings*. The server can run locally in the same computer where QGIS is running, or remotely. For instance, it is possible to connect to a remote cluster, to perform jobs that are highly demanding in terms of computational resources.

⁹ <https://github.com/gem/oq-engine>

¹⁰ <https://github.com/gem/oq-engine/blob/master/doc/web-api.md>

¹¹ <https://github.com/gem/oq-engine/blob/master/doc/running/server.md>

6.1 Run a calculation

When the *Run Calculation* button is pressed, a file explorer is opened, enabling to select the input files needed to run the job (or a zip archive containing them), including the *job.ini* file. By pressing *Open* to confirm, the job is submitted. The interface keeps querying the server asynchronously, and displaying the status of the calculation.

6.2 The list of calculations

For each calculation available on the connected OQ-Engine server (only those that the current user is authorized to visualize), the *List of calculations* (see [Dialog to run the OpenQuake Engine server](#)) shows its *Description*, its unique *Job ID*, its *Calculation Mode*, its *Owner* and its *Status* (indicating if the calculation is still running, if it failed or if it was successfully completed). The list is refreshed at regular intervals of few seconds and it displays the most recent 100 calculations.

Note: Before OpenQuake Engine 2.6, the *Job Type* was displayed instead of the *Calculation Mode*, specifying only if a calculation was classified as *hazard* or *risk*. If the plugin attempts to connect to an outdated OpenQuake Engine Server that is older than version 2.6, the column *Calculation Mode* will display the value *unknown* for all calculations. In case it happens, we recommend to update the OQ Engine Server or to downgrade the plugin to an earlier version, in order to ensure maximum compatibility between the two software versions.

It is possible to scroll the list up and down and to resize it with respect to the *List of outputs*. When any row of the list of calculations is selected, the row is highlighted and the list of outputs for the corresponding calculation is populated (it will be an empty list if the calculation has not been successfully completed). While a row is selected, it is automatically centered in the visible area of the list, at each refresh. By clicking again on the currently selected row, it will be unhighlighted, resetting the default scrolling behavior of the table.

6.3 Watch the console log

It is possible to watch the log of a calculation, by pressing the *Console* button in the corresponding row. The log will be displayed in a separate window as shown in [Example of console output for a OQ-Engine calculation](#).

6.4 Download outputs

If a calculation was completed successfully, an *Outputs* button is provided. When it is clicked, the list of available outputs is shown, which depends on the type of artifacts generated by the calculation. Outputs can be downloaded in one of the available formats, by clicking the corresponding button. In some cases, they can also be loaded as QGIS layers (whenever a corresponding *Load layer* button is available), and a default styling is applied, based on parameters chosen by the user. In some cases (e.g., damage by asset), while loading the layer, it is also possible to aggregate points by zone. In order to do so, it is necessary to specify another available layer, with polygons geospatially containing the points to be collected. The aggregation is performed leveraging an algorithm provided by the QGIS processing toolbox, called “Join attributes by location (summary)”.

```

2017-05-23T19:57:29.14,INFO,MainProcess/24900,Read 528 hazard site(s)
2017-05-23T19:57:29.18,INFO,MainProcess/24900,Instantiating the source-sites filter
2017-05-23T19:57:29.25,INFO,MainProcess/24900,Parsing /tmp/tmp1xqst/home/paolo/projects/oq-
engine/demos/hazard/AreaSourceClassicalPSHA/source_model.xml
2017-05-23T19:57:29.36,INFO,MainProcess/24900,Processed source model 1 with 1 potential gsim path(s) and 1 sources
2017-05-23T19:57:29.41,INFO,MainProcess/24900,Filtering composite source model
2017-05-23T19:57:29.47,INFO,MainProcess/24900,Using a maxweight of 200
2017-05-23T19:57:29.52,INFO,MainProcess/24900,Sending source group #1 of 1 (Active Shallow Crust, 1 sources)
2017-05-23T19:57:29.56,INFO,MainProcess/24900,Submitting "classical" tasks
2017-05-23T19:57:29.69,INFO,MainProcess/24900,Sent 20.1 KB of data in 1 task(s)
2017-05-23T19:58:09.68,INFO,MainProcess/24900,classical 100%
2017-05-23T19:58:09.74,INFO,MainProcess/24900,Received 861.23 KB of data, maximum per task 861.23 KB
2017-05-23T19:58:09.83,INFO,MainProcess/24900,Submitting "build hcurves and stats" tasks
2017-05-23T19:58:09.95,INFO,MainProcess/24900,Sent 862.33 KB of data in 4 task(s)
2017-05-23T19:58:10.00,INFO,MainProcess/24900,build hcurves and stats 25%
2017-05-23T19:58:10.05,INFO,MainProcess/24900,build hcurves and stats 50%
2017-05-23T19:58:10.10,INFO,MainProcess/24900,build hcurves and stats 75%
2017-05-23T19:58:10.16,INFO,MainProcess/24900,build hcurves and stats 100%
2017-05-23T19:58:10.21,INFO,MainProcess/24900,Received 2.41 KB of data, maximum per task 617 B
2017-05-23T19:58:10.46,INFO,MainProcess/24900,Calculation 176 finished correctly in 41 seconds

```

Fig. 6.2: Example of console output for a OQ-Engine calculation

Warning: The layers that are automatically imported from OQ-Engine outputs are stored *in memory*, unless they are explicitly saved by the user into a persistent format, such as a shapefile. Please be aware that the contents of *in memory* layers will not be saved when the QGIS project that contains them is saved. Furthermore, saving an *in memory* layer into a persistent file, only the contents of the layer will be saved, and not its style. If you want to save also the style, you have to right-click on the original layer and select *Styles -> Copy Style*, and then do the same on the target persistent layer and select *Styles -> Paste Style*.

A hazard map defines the geographic distribution of the values of a scalar IMT (see also [Definitions and acronyms](#)) characterized by a fixed probability of being exceeded at least once in a time span T . For computing a hazard map with the OpenQuake-engine the user defines an IMT, a region over which the map will be computed and, the corresponding probability of exceedance. An alternative representation of a hazard map fixes the value of IMT and computes over an area the corresponding probability of at least on exceedance in T ; this typology of result is currently not supported by the OpenQuake-engine.

A loss map provides the estimated losses for a collection of assets, for a certain return period (or probability of exceedance within a given interval of time). It is important to understand that these maps are not providing the distribution of losses for a seismic event or level of ground motion with the chosen return period, nor can the losses shown on the map be summed to obtain the corresponding aggregate loss with the same return period. This type of maps is simply providing the expected loss for a specified frequency of occurrence (or return period), for each asset.

Note: All layers created loading data from OQ-Engine outputs contain metadata that can be visualized within the “Layer Properties” panel. The “Title” of the calculation can be found in the “Identification” tab. Dozens of other parameters that are specific for each calculation type are stored in the “Keywords” tab.

Other kinds of outputs (e.g. hazard curves and uniform hazard spectra) can be visualized in the OpenQuake IRMT Data Viewer (see [The OpenQuake IRMT Data Viewer](#)).

It is also possible to download the HDF5 datastore corresponding to a chosen calculation. When the *Outputs* button is pressed, the *Download HDF5 datastore for calculation N* button is enabled (where N is the calculation ID). By pressing it and selecting a destination folder, the file is downloaded and its full path is displayed both in the QGIS message bar and in the *Log Messages Panel*.

Another button that becomes available when a calculation identified as N is selected is *Show parameters*

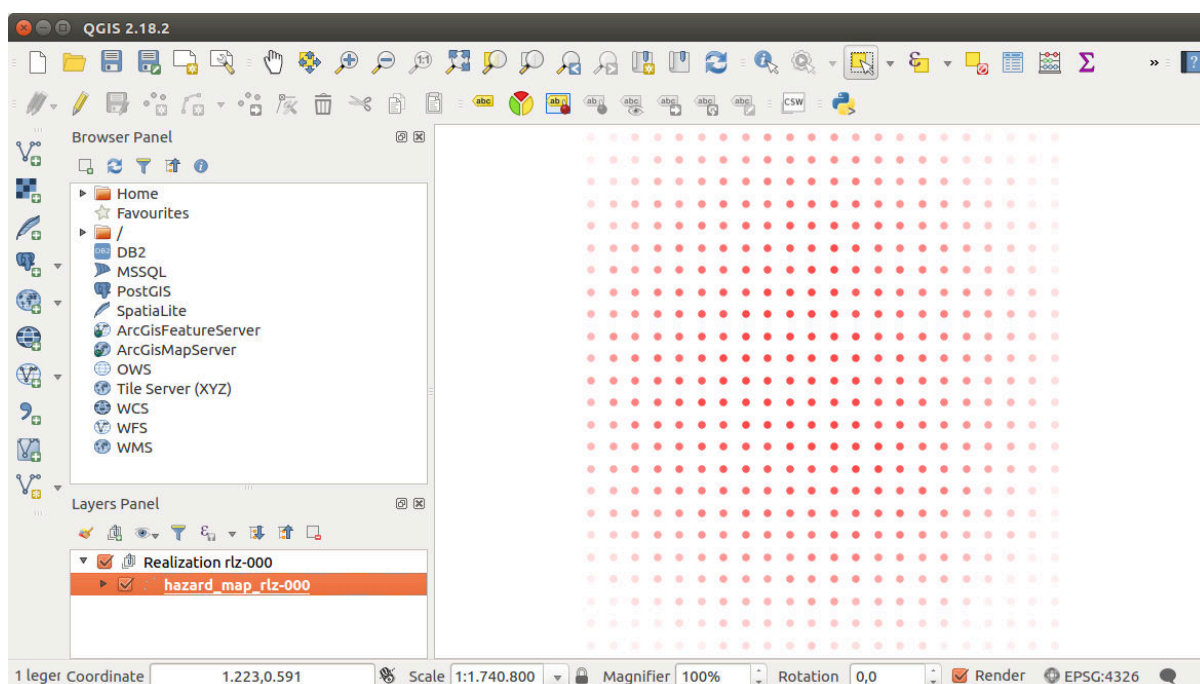


Fig. 6.3: Example of a hazard map produced by the OpenQuake Engine

of calculation *N*. It opens a dialog window with a text area displaying the parameters of the calculation (see [Example showing parameters of a OQ-Engine calculation](#)).

6.5 Run a postprocessing (or risk) calculation on top of a previously computed hazard

To run a postprocessing or risk calculation on top of a previous hazard one you have to click the *Continue* button on the corresponding hazard calculation and select the files including the postprocessing or risk *job.ini* and the related inputs (or a zip archive containing them).

6.6 Remove a calculation

To remove a calculation, press the *Remove* button. It will remove the calculation from the list, from the database and from the datastore.

6.7 Abort a calculation

To abort a calculation while it is still running, press the *Abort* button. The aborted calculation will still be visible in the list, and it will be possible to visualize the corresponding console log. Afterwards, it will be possible to completely remove the calculation from the list, from the database and from the datastore, by pressing the button *Remove*.



Fig. 6.4: Example showing parameters of a OQ-Engine calculation

6.8 Calculation status

The *Status* column of the *List of calculations* indicates the current status of a calculation, which can be one of the following: *created*, *executing*, *complete*, *failed*.

The reason behind a *failed* calculation can be inspected by watching the console log, by means of the *Console* button.

THE OPENQUAKE IRMT DATA VIEWER

The OpenQuake IRMT Data Viewer is a dock window added to QGIS by the OpenQuake IRMT plugin, used for data visualization. It is shown/hidden by pressing the button *Toggle OpenQuake IRMT Data Viewer*.

In its initial state, the window displays a *Output Type* selector, that enables to trigger the visualization of different types of data, and an initially empty plotting area.

The viewer can plot some of the outputs produced by the OpenQuake Engine, such as hazard curves and uniform hazard spectra (see also *Definitions and acronyms*).

When a layer containing compatible data is activated in the QGIS and the corresponding output type is selected, the viewer is ready to visualize the outputs corresponding to the features that will be selected in the map.

Plots are obtained using the *Matplotlib* library. Below the plotting area, Matplotlib provides a toolbox with standard functionalities that enable modifying markers, labels, axes, zooming level and other parameters, saving the plot to file and exporting the selected curves into a csv format.

7.1 Visualizing outputs of hazard calculations

This section describes how to drive the user interface of the plugin to visualize in the IRMT Data Viewer some of the hazard outputs produced by OpenQuake Engine calculations. For an extensive explanation of those outputs, please refer to the *user manual of the OpenQuake Engine*¹². In *Driving the OpenQuake Engine (Example of a hazard map produced by the OpenQuake Engine)* you can see an example of a hazard map loaded by the plugin into a QGIS vector layer. In that case the IRMT Data Viewer is not needed to display the data. Other kinds of outputs, such as hazard curves, can not be displayed in a map as QGIS layers. The IRMT Data Viewer has the purpose of displaying multi-dimensional data in an interactive way, linking the selection of sites in the map canvas to the plotting of corresponding curves.

¹² <https://docs.openquake.org/oq-engine/stable/>

7.1.1 Visualizing hazard curves

A hazard curve defines the relation between a scalar IML and the probability of at least one exceedance of that IML within a time span T . The OpenQuake-engine computes discrete hazard curves described by a two-dimensional array containing a first column of n values of an IMT and a second column including values of the probability of exceedance of the resultant IML in the time span T (which is indicated in the OpenQuake-engine configuration file). A hazard curve is the primary result of a PSHA analysis for a particular site. From hazard curves it is possible to compute other result-typologies such as hazard maps and uniform hazard spectra.

As described in *Driving the OpenQuake Engine*, the plugin enables to run hazard calculations and to download the corresponding outputs. For outputs of type *hcurves*, it is possible to load the data into a QGIS layer by pressing the button *Load npz as layer*. The layer will contain, for each point, the set of intensity measure levels and the corresponding values of probability of exceedance, for each of the available intensity measure types. While the layer is active, it is possible to select the *Output Type Hazard Curves*, to activate the visualization. When one or more points are selected in the map, the hazard curves for the chosen *Intensity Measure Type* are plotted together (Fig. 7.1). The legend also specifies the longitude and latitude of the points corresponding to each of the curves in the plot. By hovering on the legend items or on the curves, the corresponding points in the map are highlighted.

Warning: The highlighting effect produced by hovering with the mouse on legend items or curves, stops working correctly when a layer is loaded using the OpenLayers Plugin. Please note that, starting from QGIS 2.18, base maps can be added as layers without installing any external plugin (such as OpenLayers), but using the new core functionality *XYZ driver* instead. In order to do so, you have to open the *Browser Panel*, right-click on the *Tile Server (XYZ)* and select *New connection...*. Then, for instance, to add a connection to OpenStreetMap, you can insert into the text box the following string: `http://tile.openstreetmap.org/{z}/{x}/{y}.png`. Then press *Ok* and insert a name for the tile layer (e.g., *OpenStreetMap*). Afterwards, it is sufficient to double-click on the new item you have just created, to add OpenStreetMap to the *Layers Panel* and to visualize it in the map canvas.

7.1.2 Visualizing uniform hazard spectra

A Uniform Hazard Spectrum (UHS) is a typology of result that is site-specific – as in the case of hazard curves. A UHS defines a relationship between the period (or frequency) of a period-dependent (or frequency-dependent) IMT such as spectral acceleration and the resulting IMT value with a fixed probability of exceedance in a time span T .

The workflow to visualize uniform hazard spectra is almost the same as the one described above for visualizing hazard curves. In this case, the *Output Type* to be loaded as layer is *Uniform Hazard Spectra* (Fig. 7.2).

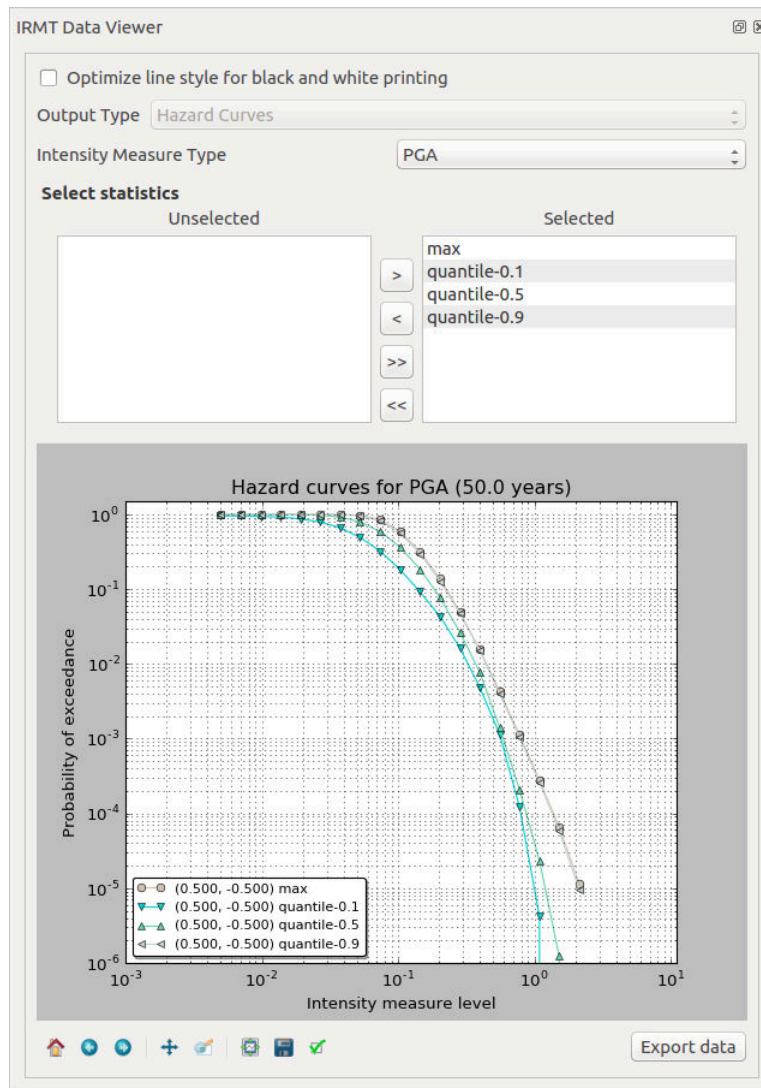


Fig. 7.1: OpenQuake IRMT Data Viewer used for displaying hazard curves

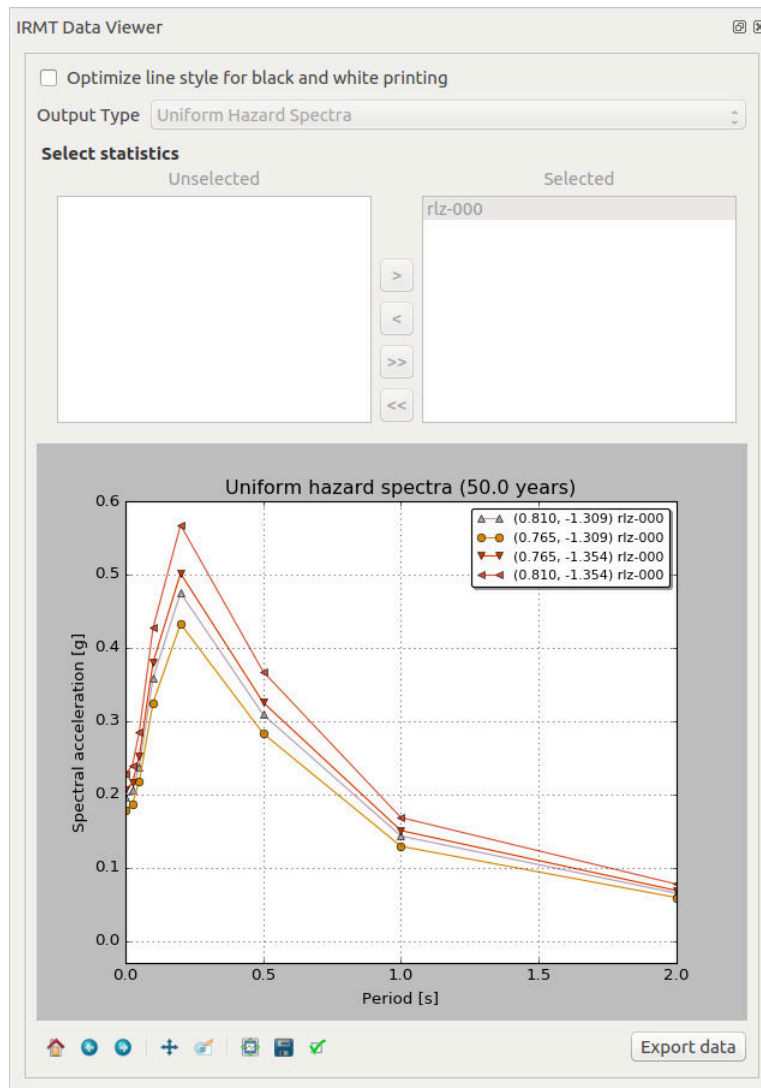


Fig. 7.2: OpenQuake IRMT Data Viewer used for displaying Uniform Hazard Spectra

7.2 Visualizing outputs of risk calculations

This section describes how to drive the user interface of the plugin to visualize some of the risk outputs produced by OpenQuake Engine calculations. For an extensive explanation of those outputs, please refer to the [user manual of the OpenQuake Engine](#)¹³.

7.2.1 Visualizing aggregate loss curves

Aggregate loss curves describe the exceedance probabilities for a set of loss values for the entire portfolio of assets defined in the exposure model.

When the button *Show* is pressed, the Data Viewer is automatically opened, providing a dropdown menu to select one of the available loss types, and a tool to select multiple realizations or statistics. By default, the first available loss type is pre-selected and all the realization or statistics are displayed in the plot. Any change in these selections produces an automatic update of the plot. The plot shows in abscissa the return period (in years) and in ordinate the aggregate loss (the measurement unit depends on the parameters of the OQ-Engine calculation).

7.2.2 Visualizing aggregate damage by asset

When the button *Aggregate* is pressed, the Data Viewer is automatically opened, providing a dropdown menu to select one of the available realizations, and another one to select one of the available loss types. Two widgets enable the selection of multiple tag names and, for each tag, one or more of its values. The additional checkbox *Exclude “no damage”* is checked by default, excluding from the plot the damage “no damage” state, which in most cases is predominant with respect to the others and therefore the corresponding bar would look too high in proportion with the others. The bar plot shows the damage distribution, where each bar corresponds to one of the damage states, and the height of the bar corresponds to the number of asset in that damage state. If no filter is selected, the whole damage distribution is displayed. If something is selected, a text field lists the selected tags, each with the chosen value. If a tag is selected, but no corresponding value is chosen, the filter will not be applied.

7.2.3 Visualizing aggregate losses by asset

When the button *Aggregate* is pressed, the Data Viewer is automatically opened, providing a dropdown menu to select one of the available loss types. Two widgets enable the selection of multiple tag names and, for each tag, one or more of its values. If the value *** is chosen, aggregate losses for each value are reported. The results are presented as a table containing, for each tag and for each realization, the corresponding aggregate loss. If no filter is selected, the total losses for each realization are displayed. If something is selected, a text field lists the selected tags, each with the chosen value. If a tag is selected, but no corresponding value is chosen, the filter will not be applied.

¹³ <https://docs.openquake.org/oq-engine/stable/>

VISUALIZING OPENQUAKE VOLCANO SCENARIO CALCULATOR OUTPUTS

The volcano scenario calculator serves as a pilot case for using OpenQuake-engine to assess risk due to hazards different than earthquakes. Detailed information for earthquake analyses is available in the [user manual of the OpenQuake Engine](#)¹⁴. A video tutorial explaining how to use the OpenQuake engine and the plugin for volcano scenario calculations is available on [YouTube](#)¹⁵.

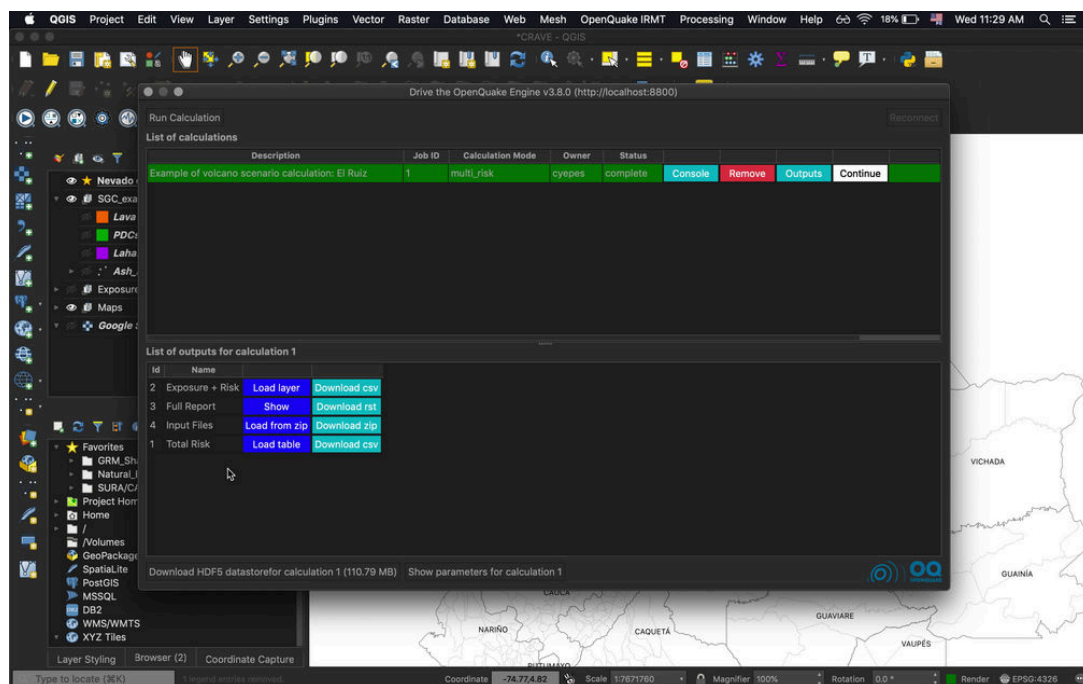


Fig. 8.1: List of outputs for a multi_risk calculation.

¹⁴ <https://docs.openquake.org/oq-engine/stable/>

¹⁵ <https://www.youtube.com/watch?v=I2KEHUAZ-j8>

8.1 Visualizing Exposure + Risk

By pressing the *Load layer* button corresponding to the *Exposure + Risk* row in the list of outputs for the *multi_risk* calculation, the dialog in figures *Dialog to select parameters for the visualization of exposure.* and *Dialog to select parameters for the visualization of risk.* will be displayed:

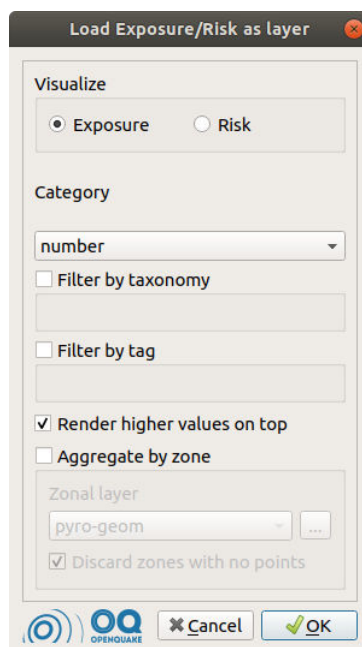


Fig. 8.2: Dialog to select parameters for the visualization of exposure.

Selecting the corresponding radio button, it is possible to visualize either the exposure or the risk. In this example, the exposure can be considered in terms of number of buildings or occupants, or in terms of structural value. For what concerns the risk, it is possible to select the kind of peril to take into account (in this case: dry or wet ash, lava, lahars or pyroclastic flow) and the category of interest (e.g.: structural collapse, loss or no-damage).

Both for exposure and risk, it is possible to filter the available data by taxonomy and/or by tag. One or multiple items of interest can be chosen (multiple taxonomies and/or multiple values per one of the available tags).

The checkbox *Render higher values on top* is checked by default, as it is recommended. It prevents features with smaller values from overlapping more interesting, higher-valued features that might be located at the same coordinates or very close to them.

Once all the parameters above are set, it is possible to load the corresponding data as a QGIS layer, by pressing the *OK* button. If the *Aggregate by zone* checkbox is not selected, the resulting layer will contain points at the coordinates for which the output data has been calculated. It is also possible to press the *OK* button after checking *Aggregate by zone* and selecting a layer containing polygonal geometries (e.g., each polygon may represent the boundaries of a different municipality). In this case, a geospatial aggregation is performed, summing the values of points contained in each polygon.

In both cases, with or without zonal aggregation, the layer that is produced will be styled automatically.

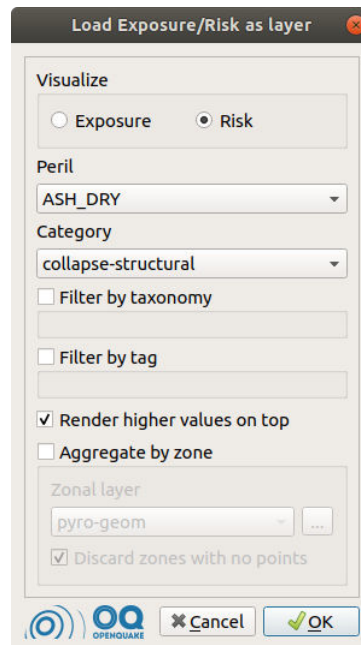


Fig. 8.3: Dialog to select parameters for the visualization of risk.

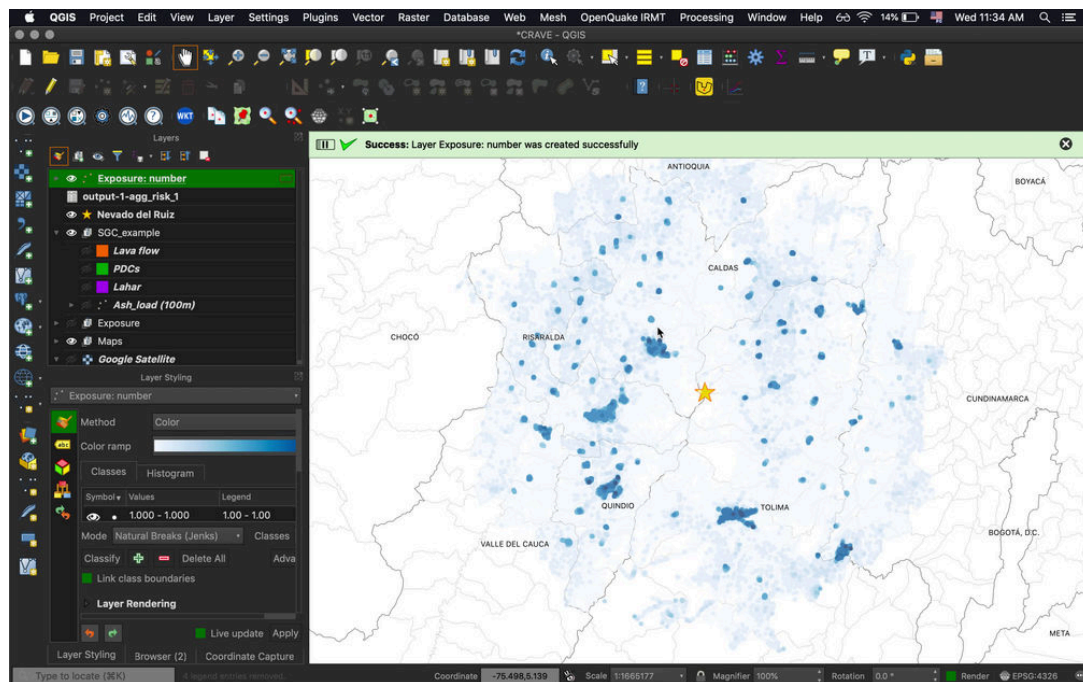


Fig. 8.4: An example of exposure map in terms of number of buildings.

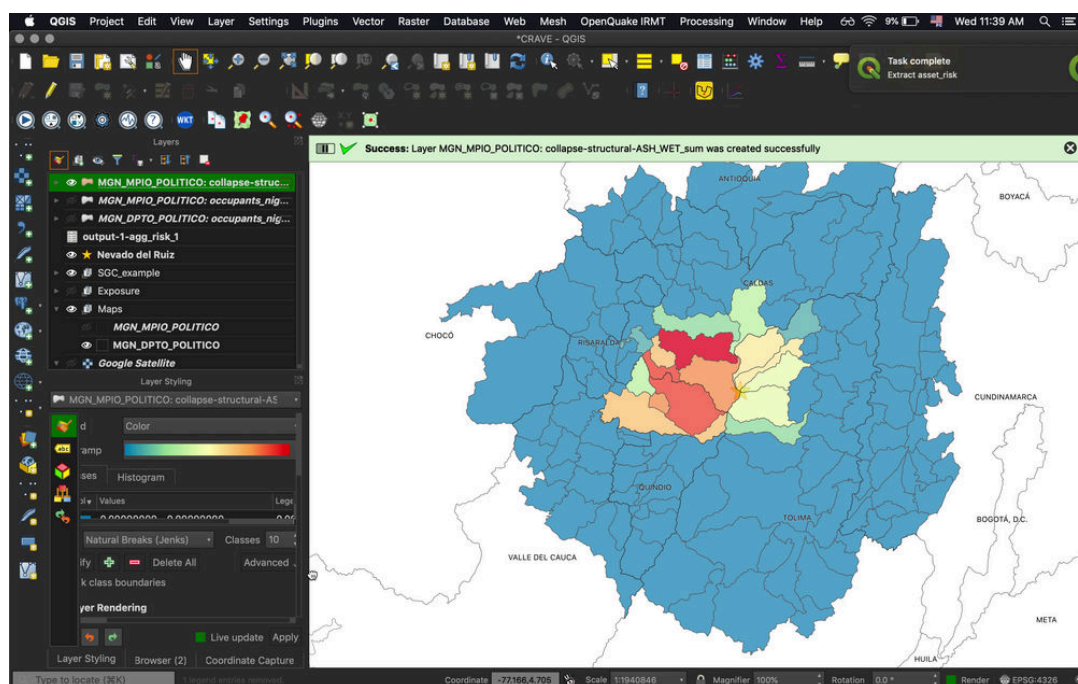


Fig. 8.5: An example displaying the estimated number of buildings per region that would collapse due to wet ash fall.

8.2 Visualizing Input Files

By pressing *Load from zip*, the following dialog is displayed.

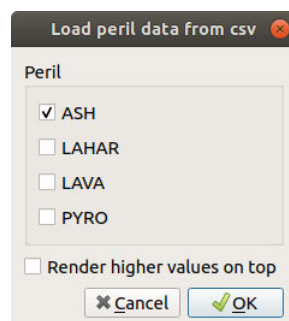


Fig. 8.6: Dialog to select the input files corresponding to the footprints of the perils to be visualized.

It allows to visualize the input data corresponding to the selected perils. Also in this case, it is possible to check *Render higher values on top*, in order to avoid hiding interesting high-valued points below other rendered points.

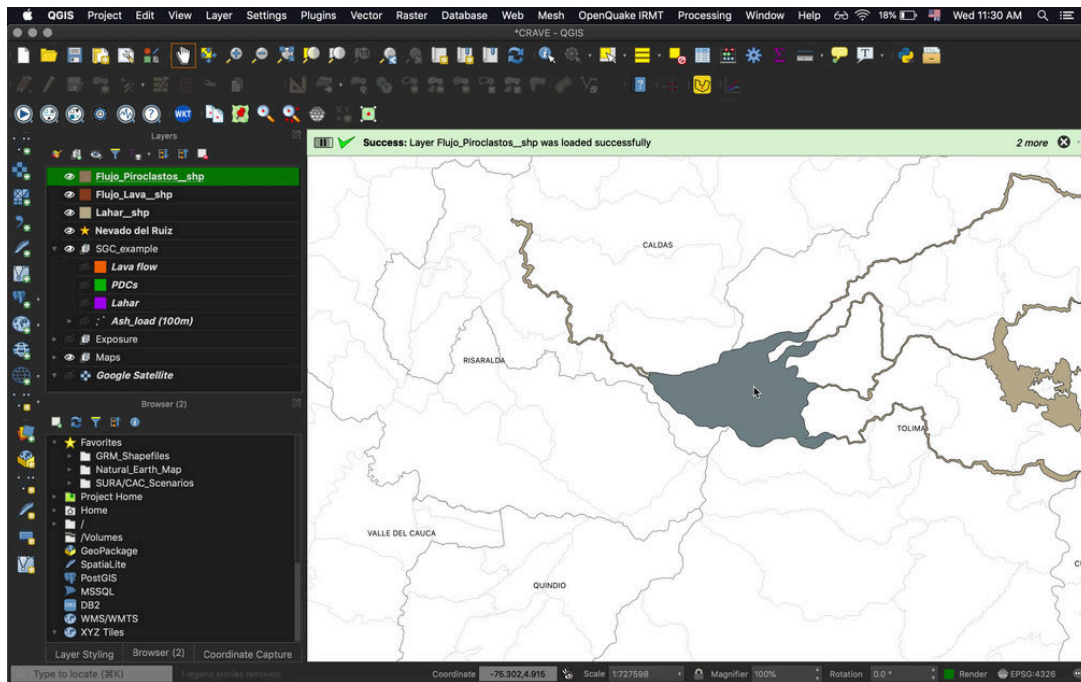


Fig. 8.7: An example of volcano footprint input.

8.3 Visualizing Total Risk

By pressing *Load table*, the plugin will display a summary of the total risk for each of the analyzed perils, with respect to each category of risk and exposure.

| | peril | number | ccupants_high | loss-structural | damage-struct | llapse-structu |
|---|---------|--------|---------------|-----------------|---------------|----------------|
| 1 | PYRO | 4 | 11 | 40000 | nan | nan |
| 2 | LAVA | 1 | 1 | 20000 | nan | nan |
| 3 | LAHAR | 2 | 10 | 40000 | nan | nan |
| 4 | ASH_WET | nan | nan | 6804.35 | 171.337 | 2.66304 |
| 5 | ASH_DRY | nan | nan | 1861.9 | 171.955 | 2.04524 |

Show All Features

Fig. 8.8: Table summarizing the total risk.

Note: The style that is assigned automatically to the layers might need to be slightly modified on a case-by-case basis, for instance modifying the classification criteria, e.g. changing the number of classes and/or the classification mode (using equal intervals, quantiles, pretty breaks, etc.). A convenient way to do so is to use the “Layer styling” panel and select one of the available layers to visualize and modify its styling settings. The same can be obtained by double clicking one of the listed layer names, thus opening the layer properties dialog, and using the functionalities offered by the “Symbology” tab.

AGGREGATING POINTS BY ZONE

In previous versions of this plugin, there was a specific tool to perform the geospatial aggregation. Recently, it was dismissed because its functionalities overlap with those of a powerful algorithm available in the QGIS Processing Toolbox. Therefore, the button that pointed to our customized implementation now points to the corresponding Processing tool, called “Join attributes by location (summary)”. Such button, *Aggregate points by zone*, is under the *Utilities* submenu of the *OpenQuake IRMT* menu.

In order to obtain a similar workflow with respect to what was done previously with our tool, we suggest to select the *zonal layer* as “Input layer” and the *points layer* as “Join layer”, then keep checked “intersects” as “Geometric predicate”, and choose the fields to summarize and the summaries to calculate (that in our case were “sum” and “mean”). When the plugin opens the user interface of the algorithm, it presets the above configuration by default.

Afterwards, it is possible to style the layer using the standard QGIS layer symbology menu, for instance setting a “Graduated” renderer by a chosen column and applying the classification by a chosen number of classes. In our standard workflow, the column to be chosen is the “_sum” field that was created by the aggregation tool.

TRANSFORMING ATTRIBUTES

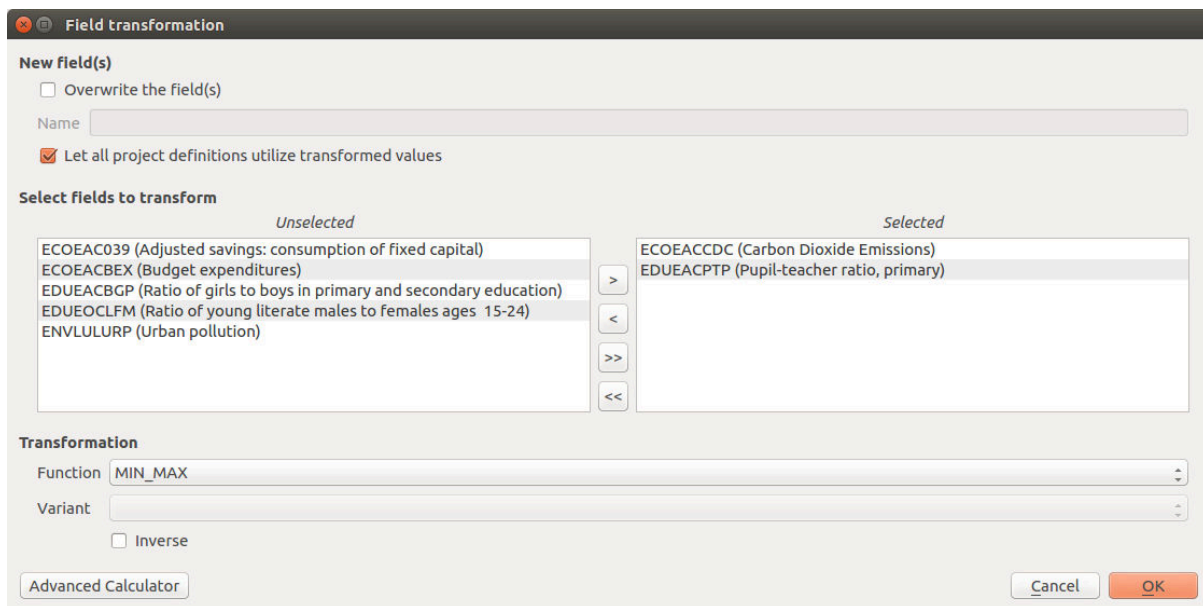



Fig. 10.1:  Variable transformation and batch transformation functionality

When variables are defined in incommensurate ranges or scales, they can be standardized to avoid problems inherent when mixing measurement units, and normalization is employed to avoid having extreme values dominate an indicator, and to partially correct for data quality problems. The QGIS platform natively provides a *Field calculator* that can be used to update existing fields, or to create new ones, in order to perform a wide variety of mathematical operations for the standardization/transformation of data. In addition, the OpenQuake IRMT provides a number of transformation functions found in popular statistical and mathematical modelling packages (Table 10.1).

Table 10.1: Selection of transformation functions with equations found in the OpenQuake IRMT.

| | |
|-------------------------------|--|
| Standardization (or Z-scores) | $Z(x_i) = \frac{x_i - \mu_x}{\sigma_x} \quad (\mu_x = \text{mean} \quad \sigma_x = \text{stddev})$ |
| Min-Max | $M(x_i) = \frac{x_i - \min_{i \in \{1, \dots, n\}}(x_i)}{\max_{i \in \{1, \dots, n\}}(x_i) - \min_{i \in \{1, \dots, n\}}(x_i)}$ |
| Logistic Sigmoid | $S(x_i) = \frac{1}{1 + e^{-x_i}}$ |
| Simple Quadratic | $Q(x_i) = \frac{x_i^2}{\max_{i \in \{1, \dots, n\}}(x_i)}$ |

Warning: Not all layer types can be edited. For instance, it is impossible to add or modify fields of a csv-based layer. Prior to apply transformations to non-editable layers, it is necessary to save them as shapefiles or as another editable kind.

These include:

1. **Ranking** is the simplest standardization technique. It is not affected by outliers and allows the performance of enumeration units to be benchmarked over time in terms of their relative positions (rankings). The ranking algorithm deals with ties using a chosen strategy between those listed below (see https://en.wikipedia.org/wiki/Ranking#Strategies_for_assigning_rankings):
 - Average - Fractional (1 2.5 2.5 4)
 - Standard competition - Minimum (1 2 2 4)
 - Modified competition - Maximum (1 3 3 4)
 - Dense (1 2 2 3)
 - Ordinal (1 2 3 4)
2. **Z-scores normalization** is the most common standardization technique. A Z-score converts indicators to a common scale with a mean of zero and standard deviation of one. Indicators with outliers that are extreme values may have a greater effect on the composite indicator. The latter may not be desirable if the intention is to support compensability where a deficit in one variable can be offset (or compensated) by a surplus in another. Summarizing: Z-score handles outliers, but does not produce normalized data with the exact same scale.
3. **Min-Max normalization** is a type of transformation that rescales variables into an identical range between 0 and 1. Extreme values/or outliers could distort the transformed risk indicator. However, the Min-Max transformation can widen a range of indicators lying within a small interval, increasing the effect of the variable on the composite indicator more than the Z-scores. In brief: Min-Max guarantees all normalized fields will have the exact same scale but does not handle outliers well.
4. **Log10** is one of a class of logarithmic transformations that include natural log, log2, log3, log4, etc. Within the current plugin, we offer functionality for log10 only, yet these transformations are possible within the advanced *field calculator*. A logarithm of any negative number or zero is undefined. This algorithm copes with the case in which any zeros are present in the input data, offering two strategies:
 - Ignore zeros: each zero in input is transformed into nan
 - Increment all values by one: each input value is incremented by 1 before running the transformation.

The algorithm uses the `numpy.log10` function to transform the (possibly modified) list of values.

5. **Sigmoid function** is a transformation function having an S shape (sigmoid curve). A Sigmoid function is used to transform values on $(-\infty, \infty)$ into numbers on $(0, 1)$. The Sigmoid function is often utilized because the transformation is relative to a convergence upon an upper limit as defined by the S-curve. The OpenQuake IRMT utilizes a *simple sigmoid function* as well as its inverse. The Inverse of the Sigmoid function is a logit function which transfers variables on $(0, 1)$ into a new variable on $(-\infty, \infty)$.
6. **Quadratic or U-shaped functions** are the product of a polynomial equation of degree 2. In a quadratic function, the variable is always squared resulting in a parabola or U-shaped curve. The

OpenQuake IRMT offers an increasing or decreasing variant of the quadratic equation for horizontal translations and the respective inverses of the two for vertical translations.

Note: It may be desirable to visualize the results of the application of transformation functions to data. Although not feasible within the plugin at this point, we intend to build data plotting and curve manipulating functionalities into future versions of the toolkit.

The *Transform attribute* dialog (Fig. 10.1) was designed to be quite straightforward. The user is required to select one or more numeric fields (variables) available in the active layer. For the selection to be completed, the user must move the variables (either one at a time, or in a batch) to the *Selected variables* window on the right side of the interface. The user must then select the function necessary to transform the selected variables. For some functions, more than one variant is available. For functions that have an implementation of an inverse transformation, the *Inverse* checkbox will be enabled to allow users to invert the outcome of the transformation.

The *New field(s)* section contains two checkboxes and a text field. If the first checkbox *Overwrite the field(s)* is selected, the original values of the transformed fields will be overwritten by the results of the calculations; otherwise, a new field for each transformed variable will be created to store the results. In situations in which a user may desire to transform variables one at a time rather than using a batch transformation process, it is possible for the user to name each respective new field (editing the default one proposed by the tool). Otherwise, the names of the new fields will be automatically assigned using the following convention: if the original attribute is named *ORIGINALNA*, the name of the transformed attribute becomes *_ORIGINALN* (prepending “_” and truncating to 10 characters which is the maximum length permitted for field names in shapefiles). If the layer does not have the limitations of a shapefile, the name of the transformed field will *not* be truncated to 10 characters.

Note: In the lists of fields, both the field name and the field alias are displayed, with the format *name (alias)*. If no alias is specified for the field, the parenthesis will be empty. The plugin automatically assigns to the transformed field the same alias of the original one (if available). Please make sure that the names of the fields to be transformed do not contain parentheses, otherwise the plugin would erroneously interpret them as containers for the alias; therefore the selected name would be incomplete (being taken excluding the parentheses) and it would not be found in the layer.

By clicking the *Advanced Calculator* button, the native QGIS field calculator is opened. Please refer to the [code documentation](#) for the detailed description of all the algorithms and variants provided by the OpenQuake IRMT.

The plugin is also an algorithm provider (the *OpenQuake IRMT* provider) for the Processing Toolbox. The transformation functions described above are available under the *Field transformation* group.

LEVERAGING R TO PERFORM STATISTICAL ANALYSIS IN QGIS

Socioeconomic analysis often involve statistical investigation, for instance for dimensionality reduction in problems involving a considerable number of variables. QGIS is well integrated with tools that can be leveraged for this kind of purpose. R¹⁶ is a well-known free software environment for statistical computing and graphics. It is widely used and easily installable on all the most used operating systems. QGIS enables users to drive R scripts from within the QGIS Processing Toolbox¹⁷. This documentation¹⁸ explains how to make R scripts available in the Processing Toolbox. You can create your own scripts or download examples from a public repository¹⁹. Scripts can be managed and edited directly within QGIS, through an embedded graphical widget. This feature is especially convenient to users who need to quickly customize existing scripts. However, it does not provide the same flexibility and interactivity that the R console offers, therefore the coding process becomes a little more challenging. By default, on unix-based systems, scripts are locally collected in the directory `~/.qgis2/processing/rscripts` and they are identified by a file like `scriptname.rsx`, that contains the script itself, and an optional file like `scriptname.rsx.help`, that provides the corresponding documentation. An example taken from the above public repository is the following:

File *Summary_statistics.rsx*:

```
##Basic statistics=group
##Layer=vector
##Field=Field Layer
Summary_statistics<-data.frame(rbind(sum(Layer[[Field]]),
length(Layer[[Field]]),
length(unique(Layer[[Field]])),
min(Layer[[Field]]),
max(Layer[[Field]]),
max(Layer[[Field]])-min(Layer[[Field]]),
mean(Layer[[Field]]),
median(Layer[[Field]]),
sd(Layer[[Field]])),row.names=c("Sum:", "Count:", "Unique values:", "Minimum_
↪value:", "Maximum value:", "Range:", "Mean value:", "Median value:", "Standard_
↪deviation:"))
colnames(Summary_statistics)<-c(Field)
>Summary_statistics
```

File *Summary_statistics.rsx.help*:

¹⁶ <https://www.r-project.org/>

¹⁷ http://docs.qgis.org/2.0/en/docs/user_manual/processing/toolbox.html

¹⁸ https://docs.qgis.org/testing/en/docs/training_manual/processing/r_intro.html

¹⁹ <https://github.com/qgis/QGIS-Processing/tree/master/rscripts>


```
{
  "ALG_DESC": "This tool calculates the following summary statistics for a
              numeric field: (1) Sum, (2) Count, (3) Unique values,
              (4) Minimum value, (5) Maximum value, (6) Range, (7) Mean,
              (8) Median and (9) Standard deviation.\n\n",
  "R_CONSOLE_OUTPUT": "Summary statistics table",
  "ALG_CREATOR": "Filipe S. Dias, filipesdias(at)gmail.com",
  "Layer": "Input vector with at least one numeric field",
  "Field": "Numeric field",
  "ALG_HELP_CREATOR": "Filipe S. Dias, filipesdias(at)gmail.com"}
}
```

Note that the header of the script contains some lines beginning with `##`. These lines are used by the QGIS Processing Toolkit to build a graphical user interface that will be displayed every time the script is launched, and that enables users to graphically set up some parameters that will be used by the script. In the example, for instance, a dropdown menu will list all the available vector layers; once a layer is selected, another dropdown menu will list all its fields. Please refer to [the QGIS documentation](#)²⁰ for a more detailed description of the script syntax.

A brief list of statistical techniques that are often needed in socioeconomic analysis, that are all available in R, are as follows:

- Summary statistics (mean, median, high value, low value, standard deviation)
- Boxplots as well as measures of skewness and kurtosis
- Histograms
- Scatter plotting
- Correlation (Pearson's R, Spearman Rank, Kendall's Tau)
- Cronbach's Alpha (which is based on correlation)
- Principal Components Analysis/Factor Analysis

Some of these require the statistical platform to take into account one single field (e.g., summary statistics) or a couple of fields (e.g. scatterplot). In such cases, it is sufficient to add parametric references to those fields into the header of the script. Things become more complicated where the number of parameters required is not known in advance, as in the case of Principal Components Analysis. In older versions of QGIS, [the widgets that can be used through the script header syntax](#)²¹ did not include a multiselection functionality. Therefore, it was impossible for the user to graphically select an indefinite number of fields from the complete list. However, even without a multiselect widget, it was possible to let a script perform the analysis on the whole set of numeric fields available in the selected layer. In order to exclude some numeric fields from the analysis, we can add to the script header the reference to a textual field to be added to the GUI, in which the user can insert a list of comma-separated names of fields to be ignored. The script becomes something like:

```
##Basic statistics=group
##Layer=vector
##Exclude=String
layerData <- data.frame(Layer)
numericFields <- sapply(layerData, is.numeric)
numericData <- layerData[, numericFields]
```

(continues on next page)

²⁰ http://docs.qgis.org/2.8/en/docs/user_manual/processing/3rdParty.html#r-creating-r-scripts

²¹ http://docs.qgis.org/2.0/en/docs/user_manual/processing/modeler.html#definition-of-inputs

(continued from previous page)

```
excludedFields <- trimws(strsplit(Exclude, ",")[[1]])
analyzedData <- numericData[, !(names(numericData) %in% excludedFields)]
comps <- prcomp(analyzedData, scale=TRUE)
>comps
```

Since a multiple selector for layer fields has been made available, it is possible to run R algorithms such as in the following example, using the *multiple field* widget type:

```
##Basic statistics=group
##Layer=vector
##ChosenFields=multiple field Layer
layerData <- data.frame(Layer)
numericFields <- sapply(layerData, is.numeric)
numericData <- layerData[, numericFields]
chosen <- trimws(strsplit(ChosenFields, ",")[[1]])
analyzedData <- numericData[, names(numericData) %in% chosen]
comps <- prcomp(analyzedData, scale=TRUE)
>comps
```

This script creates automatically a graphical user interface that lets the user select one of the available layers. Once the layer is selected, the corresponding fields are listed in a multi-select widget, where a set of them can be chosen. As soon as the *Run* button is pressed, the algorithm collects the chosen fields and it performs the Principal Components Analysis on them, excluding the non-numeric ones that might have been erroneously selected.

What if we want to build a vector layer using the results of a calculation performed by R? Examples of this are available in the documentation linked above. The following example shows how to load a vector layer from those available in QGIS, make a copy of it, perform a calculation, save the result in a new field of the new layer, and make the final layer available in QGIS. In this basic example, the calculation is extremely simple, just summing the values of two fields. However, it can be easily extended to obtain complex results.

```
##Vector processing=group
##Layer=vector
##First=Field Layer
##Second=Field Layer
##output=output vector
modified <- data.frame(Layer)
modified['SUM'] <- NA
modified$SUM <- Layer[[First]] + Layer[[Second]]
output=SpatialPolygonsDataFrame(Layer, as.data.frame(modified))
```

We have just seen in the latter example that the output of a script can be a vector layer (*##output=output vector*). In the previous example, *>comps* indicated to the Processing Toolkit that the object *comps* had to be shown to the user as text in a dedicated output window. A third possibility is to display the output as a plot, using the *##showplots* directive as follows:

```
##Vector processing=group
##showplots
##Layer=vector
##Field=Field Layer
```

(continues on next page)

(continued from previous page)

```
##Unit=String  
boxplot(Layer[[Field]], main="BOXPLOT", xlab=paste(Field), ylab=paste(Unit))
```

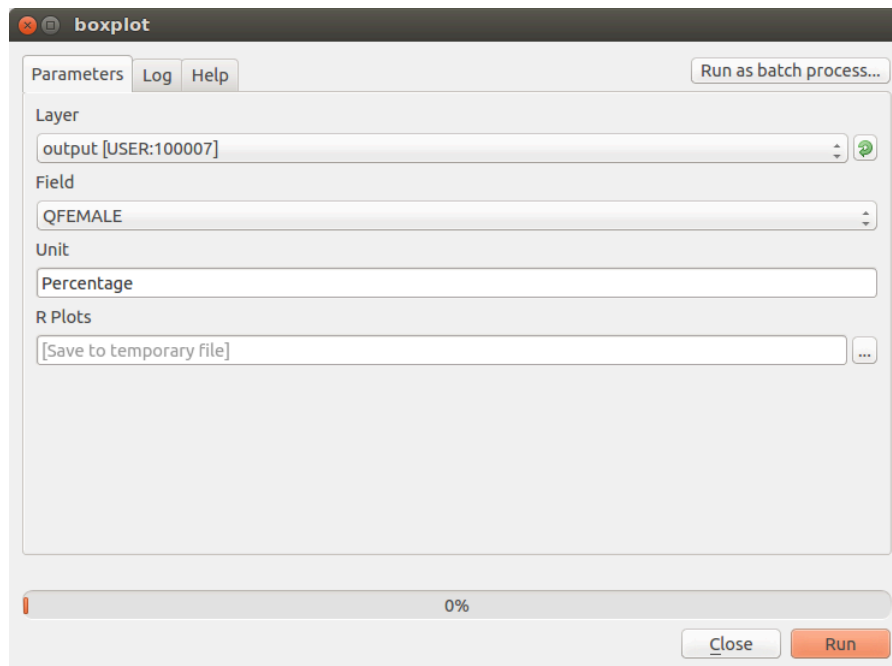


Fig. 11.1: Boxplot Dialog Window

When the script is executed, the dialog shown in Fig. 11.1 allows the user select one of the available layers, then one of its fields. In the *Unit* text field, the user can write the measurement unit to be displayed in the y label in the plot. By pressing *Run*, R is started, running the *boxplot* function on the selected data and setting the plot labels accordingly (see Fig. 11.2).

Links:

[Code repository²²](#)

[Bug tracker²³](#)

Note: This documentation is distributed within the plugin package and it is also available online at <http://docs.openquake.org/oq-irmt-qgis/>

²² <https://github.com/gem/oq-irmt-qgis>

²³ <https://github.com/gem/oq-irmt-qgis/issues>

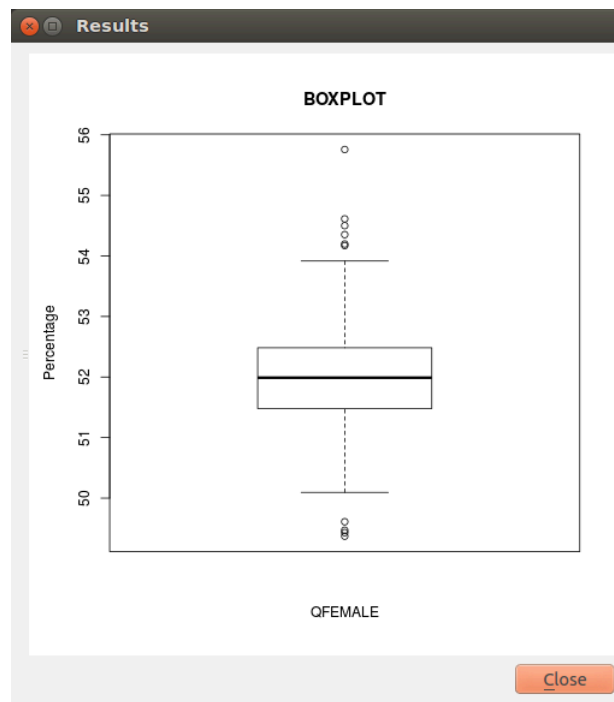


Fig. 11.2: Boxplot Output Window

BIBLIOGRAPHY

- [PMW+14] Pagani, M., Monelli, D., Weatherill, G., Danciu, L., Crowley, H., Silva, V., Henshaw, P., Butler, L., Nastasi, M., Panzeri, L., Simionato, M. and Vigano, V. OpenQuake Engine: An Open Hazard (and Risk) Software for the Global Earthquake Model. *Seismological Research Letters*, vol. 85 no. 3, 692-702
- [SCP+14] Silva, V., Crowley, H., Pagani, M., Monelli, D., and Pinho, R., 2014. Development of the OpenQuake engine, the Global Earthquake Model's open-source software for seismic risk assessment. *Natural Hazards* 72(3), 1409-1427.
- [ESRI98] ESRI Shapefile Technical Description, Environmental Systems Research Institute, Redlands, C.A.