International Cartographic Association
Association Cartographique Internationale

FRMOD, a Python tool for statistical landslide susceptibility assessment

Dávid Gerzsenyi [a],*

[a] *Institute of Cartography and Geoinformatics, Eötvös Loránd University, Hungary, 1117 Budapest, Pázmány Péter sétány 1/A, gerzsd@student.elte.hu*

* Corresponding author

Abstract:

Locating landslide-prone slopes is important, as landslides often threaten life or property where they occur. There is an abundance of statistical methods in the literature for estimating susceptibility to landslides, i.e., the likelihood of landslide occurrence based on the analyzed conditions. Still, there is a lack of readily available GIS tools for landslide susceptibility analysis, making it hard to reproduce or compare the results of different susceptibility assessments. The FRMOD is a Python-based tool for conducting landslide susceptibility analysis with the frequency ratio method. The frequency ratio method yields susceptibility estimates by comparing the frequency distributions of a set of variables from the sample landslide areas to the distributions for the whole study area. The estimates show the level of similarity to the sample landslides. The two main inputs of the tool are the raster grids of the analyzed continuous (e.g., elevation, slope) and thematic (e.g., lithology) variables and the mask grid that marks the landslide and the non-landslide areas. The analysis is performed with cross-validation to measure the predictive performance of the model. Data computed during the analysis is stored along the final susceptibility estimates and the supplementary statistics. The script reads and writes GDAL-compatible rasters, while the statistics can be saved as text files. Basic plotting functionalities for the grids and the statistics are also built-in to quicken the evaluation of the results. FRMOD enables the swift testing of different analysis setups and to apply the same analysis method for different areas with relative ease.

Keywords: landslides, landslide susceptibility, Python, frequency ratio, likelihood ratio

1. Introduction

Landslides pose threat to life and property where they interact with the human environment. The first step for minimizing the losses caused by these slope failures is locating the areas where they have occurred in the past, then where they are likely to occur in the future. Past landslides are recorded in landslide inventories, the landslide inventory can then serve as a base for landslide susceptibility studies (Parise, 2001). Landslide susceptibility maps show the relative likelihood of landslide occurrence based on the analyzed conditions, but they do not report on when exactly landslides will happen again (Brabb, 1984; Soeters and van Westen, 1996).

Methods for landslide susceptibility assessment build on the ‘past is the key to the future’ principle (Varnes 1984), i.e., that landsliding in the future will likely happen under conditions that resulted in landslides in the past. The assessment can be expert-based, where the analyst decides whether a slope is susceptible, or it can be based on statistical or physical models (Soeters and van Westen, 1996; Thiery et al., 2020).

Statistical landslide susceptibility models yield susceptibility estimates by exploring the relationship between the considered terrain and environmental factors and the distribution of past landslides in an area (Guzzetti et al., 1999). Reichenbach et al. (2018) provides an extensive review of the different statistical susceptibility assessment methods and their use in the literature. While several methods have been tried out all over the world, there is a lack of readily available GIS-tools for applying them. This leaves it to the individual analysts to find the correct way of applying a certain method. The lack of commonly agreed and well-communicated ways for applying a certain method makes it hard to compare the findings of different studies or to use them for other areas.

The frmod Python library is a tool for statistical landslide susceptibility modelling with probabilistic methods. The script performs the analysis with the frequency ratio (Lee and Talib, 2005) or the likelihood ratio (Chung, 2006) model. Both methods work with the landslide inventory of the area and a set of raster layers that describe the conditions considered during the analysis. They analyze the frequency distribution of the variables and mark those areas susceptible that are most like the sample landslides from the inventory. The scope of the program is the analysis of the already prepared input data. The results of the analysis are raster grids and supplementary statistics that can be used as base data for landslide susceptibility maps or databases.

In this paper, the theory behind the chosen analysis methods and the setup of their Python implementation is described. Guidelines for susceptibility analysis with the script and for input data preparation are also given.

Advances in Cartography and GIScience of the International Cartographic Association, 3, 2021.
30th International Cartographic Conference (ICC 2021), 14–18 December 2021, Florence, Italy. This contribution underwent double-blind peer review based on the full paper. https://doi.org/10.5194/ica-adv-3-4-2021 | © Author(s) 2021. CC BY 4.0 License

2. Susceptibility estimation method

Probabilistic landslide susceptibility assessment methods assume that landslide-affected areas in the future will have similar terrain and environmental conditions to the already landslide-affected areas (Brabb, 1984; Parise, 2001). The goal of the assessment is to find the conditions that are the most characteristic to the landslide-affected areas. Probabilistic methods usually take two types of input materials: a mask layer that marks the landslides in the study area, and a set of thematic or continuous layers, the analyzed variables.

2.1 The frequency and likelihood ratios

The frmod package can perform two types of analysis that have very similar core concepts: the frequency ratio analysis – as used by Lee and Talib (2005) – and the likelihood ratio analysis (Chung, 2006).

Frequency ratios (FR) are computed by dividing the relative frequency distribution of the analyzed variable (v) for the landslide area (LS) with the relative frequency distribution for the total study area (T).

$$FR(v) = \frac{LS(v)}{T(v)} \tag{1}$$

Likelihood ratios (LR) are computed by dividing the relative frequency distribution of the analyzed variable (v) for the landslide area (LS) with the relative frequency distribution for the non-landslide area (NLS).

$$LR(v) = \frac{LS(v)}{NLS(v)} \tag{2}$$

The total area is made up by the landslide and non-landslide areas (T = LS ∪ NLS). The two ratios are usually almost identical, because the landslide area is a few orders of magnitude smaller than the non-landslide area in most cases.

2.2 Estimating susceptibility

The rest of the workflow is the same for both ratio types. The frequency (FR) or likelihood (LR) ratios are assigned to the categories of the analyzed variable (v) layers to create the weighted layers (W). Values more characteristic to the landslide areas get weights (ratios) higher than 1, while values that are less characteristic to the landslide parts get weights less than 1. The susceptibility estimates (S) are the average of the weighted variable layers:

$$S(v1, v2, \ldots, vi) = \frac{\sum_1^i W(i)}{i} \tag{3}$$

where i is the number of analyzed variables (v).

The susceptibility estimates (S) are used to make a landslide susceptibility map. Susceptibility is usually converted to a percentile form: S values are ranked and sorted into 100 equally sized percentile categories. If a value is sorted into the Nth percentile, then N% of the study area has lower susceptibility than values in the Nth percentile. Susceptibility computed this way tells about the relative likelihood of slope failure in each areal unit (grid cells).

2.3 Evaluating the results with cross-validation

The aim of the susceptibility analysis is to estimate where landslides will likely occur in the future. As emphasized by Reichenbach et al. (2018), it is crucial to measure the reliability of these estimates. Here, k-fold cross-validation with random splits is used to evaluate the predictive performance of the model.

The landslide cells are randomly split into k number equal sized groups. One group is attached to the non-landslide cells, this is the test group. The other groups remain landslide cells, these are the training cells. The training-test group combinations are called folds. The analysis is then performed once with each fold. The prediction performance of a fold is evaluated by analyzing the distribution of the test landslide cells in the percentile susceptibility categories. The final susceptibility estimate layer is the average of the fold results.

To measure the quality of the estimates, the definition of a successful prediction must be decided. The goal here is to have as many of the test cells in the highest susceptibility categories as possible. A range of susceptibility values can be marked as "susceptible enough", then the proportion of the test cells that fall into the range can be measured to get an exact measurement for successful predictions. However, success measured with this approach is highly dependent on the choice of this cut-off range.

The shape of the susceptibility distribution of the test cells also provides information about the predictive performance of the analysis. Here, a "success rate curve" is constructed by computing the cumulative frequency distribution of the test cells (vertical axis) in the susceptibility categories (horizontal axis). The area under a success rate curve (its integral) is the smallest when the test cells are all in the highest susceptibility categories, when the estimates are the most accurate. Therefore, the smaller the area under the curve is, the more accurate the results are considered.

3. Python implementation

The frmod Python (3.7) package is a tool for performing the frequency (or likelihood) ratio analysis on georeferenced raster grids. The script reads the input rasters, computes weights with the preferred method, then yields susceptibility estimates and supplementary statistics. The rightness of the results is checked with k-fold cross validation. The produced data can be analyzed and plotted in the Python environment or exported as raster grids or tables.

The package has two modules: analysis and utils. The utils module handles the input and output of geospatial raster files with GDAL/OGR functions (GDAL/OGR contributors, 2021). The module builds on code snippets from the Python OGR/GDAL Cookbook (Erickson et al., 2013).

The analysis module is responsible for the data analysis and it also has basic plotting utilities. Data analysis is mostly done with NumPy methods (Harris et al., 2020). The statistics are stored either as NumPy arrays or pandas

DataFrames (McKinney, 2010). NumPy and pandas are common data formats for data analysis in Python and both allow the relatively easy export of the statistics to text files. The Matplotlib visualization library (Hunter, 2007) is used for creating the figures. The analysis module consists of three classes (VRaster, LandslideMask, FRAnalysis) and a few supporting functions.

This section describes the workflow of the analysis with the frmod package and gives guidelines for the analysis setup.

3.1 Data structure

The input data for the analysis are geospatial raster grids. The rasters are converted into 2D NumPy arrays upon import, then the script works with these 2D arrays in the following. These arrays inherit the shape and values of the rasters, but their geospatial information (projection, cellsize, position) is not preserved internally during the analysis. The 2D arrays with the shape of the input rasters are referred to as grids in the text and the program documentation. The geospatial information of an existing raster file is reattached to the 2D array grid when the array is exported as a raster. The analysis only works if the used rasters have the same shape and resolution. Therefore, the input data should be prepared accordingly.

3.2 Analysis setup

The model needs two types of input data: the analyzed raster layers and a mask raster layer that marks the landslide and non-landslide areas. The analyzed variables are imported as VRasters, while the mask is imported as the LandslideMask. After the data import, an FRAnalysis object is constructed from the list of the VRasters and the LandslideMask (Figure 1).

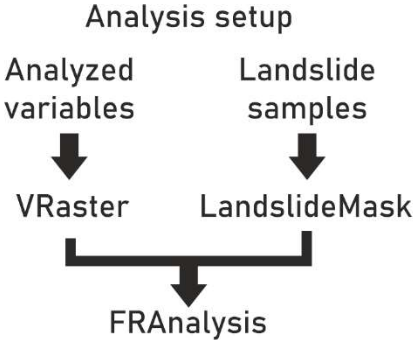

Figure 1. The workflow of the analysis setup.

3.2.1 *Analyzed variables – VRaster*

The raster layers of the analyzed variables are imported as VRaster ("variable raster") objects. VRasters can handle both continuous and categorical layers. The continuous values, where the values express a quantity, are sorted into equal width categories (bins) for the analysis. Categorical values are category IDs and they are not sorted into bins. Four parameters are needed to create a VRaster:

- name: A name for the layer.
- path: The path to the layer.
- bins: The number of bins.
- categorical: True (1) or False (0). True if the layer is categorical and False if the layer is continuous.

The values of the imported raster are stored in the VRaster's grid property as NumPy arrays. The minimum, maximum, and the nodata value of the raster are also computed when the VRaster is created. The grid (2D array) of the VRaster can be plotted with the VRaster.show method (Figure 2).

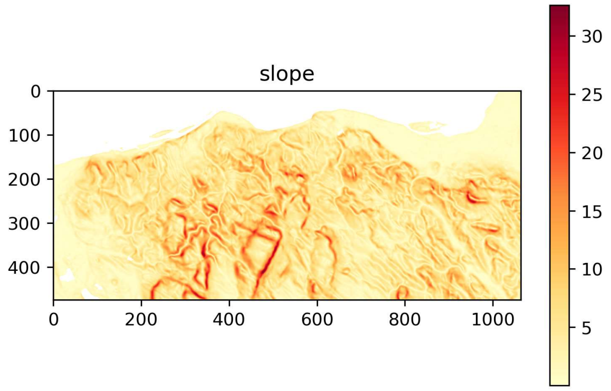

Figure 2. Example plot of a VRaster grid for slope (°) values. Created with the VRaster.show() function. Coordinates are cell positions in the plotted grid array.

3.2.2 *Mask layer – LandslideMask*

The mask layer is imported as the LandslideMask. The LandslideMask is used for delineating the study area, and its landslide and non-landslide parts. Cells outside the study area must be nodata cells, while landslide and non-landslide cells must be marked with a category ID. The folds for the cross-validation are constructed upon the creation of the LandslideMask. Five parameters are needed to create the LandslideMask object:

- name: A name for the layer.
- path: The path to the layer.
- ls_marker: The value of the landslide cells.
- nls_marker: The value of the non-landslide cells.
- fold_count: The number of cross-validation folds.

The values of the mask are stored in the grid property of the object. The folds for the cross-validation are created with the LandslideMask. The LandslideMask's train_areas property holds the grid arrays, while the indexes of the validation cells are stored in the valid_positions array.

3.2.3 *Combining inputs – FRAnalysis*

The analysis of the variable layers and the mask is done with an FRAnalysis object. It stores the analysis setup and all the data computed during the analysis. Three parameters are needed to create an FRAnalysis:

- ls_mask: The LandslideMask used as the mask.
- var_list: A list of VRasters, the list of the analyzed variables.
- classic_mode: Set True (1) to use the frequency ratio and set False (0) to use the likelihood ratio method.

3.3 Analysis

The analysis is run in three steps with three class methods of the FRAnalysis class object after its creation:

1. run_analysis: Compute the ratios and create the weighted grids for each fold and variable.
2. get_result: Create the susceptibility grids for the folds and the final susceptibility grid, compute the susceptibility distributions for the test cells.
3. get_percentile_grid: Create the percentile version of the final susceptibility array.

These class methods modify the properties of the FRAnalysis object, and they mostly work with its data.

3.3.1 *Computing ratios and weights – run_analysis*

The FRAnalysis calls its run_analysis method upon instantiation. The method computes the frequency (or likelihood) ratios for all the analyzed variables for each fold of the mask layer, then creates the corresponding weighted grids (arrays). The weighted grids are stored in the rc_folds property of the object. The statistics related to the computed weights are stored in the fr_stat_df and the fr_stats_full dictionaries. The fr_stat_df dictionary stores one table (pandas.DataFrame) with the bins and the frequency ratios of all folds. The fr_stats_full dictionary stores a set of tables for each analyzed variable, one table for each fold with the bins, computed frequency distributions and the ratios. Its keys are the VRaster names.

3.3.2 *Susceptibility estimation– get_result*

The get_result method of the FRAnalysis class works with the weighted grids (rc_folds) previously computed by the run_analysis. This method creates the susceptibility grids for the folds by averaging the weighted grids of the analyzed variables (as in Equation 3). The grids are stored in both the the raw (fold_susceptibility) and the percentile form (fold_percentiles). The final susceptibility grid (fresult) is computed here by averaging the susceptibility grids of the folds. The distribution of the validation cells in the susceptibility categories for the folds is computed after creating the susceptibility grids, and it is stored in the v_dist property.

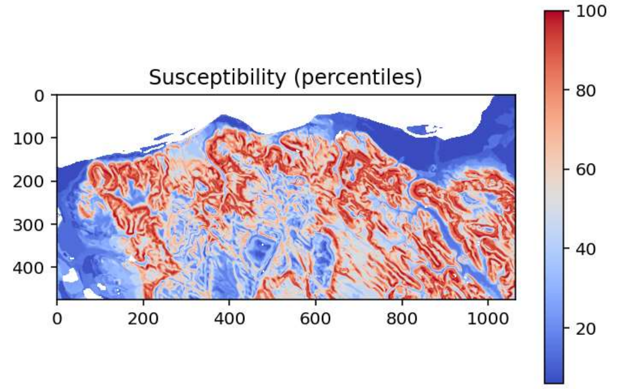

Figure 3. The estimated susceptibility in the percentile form (percentile_grid). Higher values (red) mean higher susceptibility. Coordinates are cell positions in the plotted grid array.

3.3.3 *Susceptibility map – get_percentile_grid*

The get_percentile_grid function converts the average final susceptibility grid (fresult) to the percentile form and stores it in the percentile_grid array of the FRAnalysis. Optionally, this function can also plot the percentile results if called with the show=True argument (Figure 3).

3.4 Metrics for evaluation

The evaluation of the analysis results is based on the estimated susceptibility of the test cells. This data is computed with the get_result function of the FRAnalysis and is stored in the v_dist array, as described above. The metrics for the evaluation are calculated from the susceptibility distribution of the test cells (v_dist) with the get_src and get_auc methods.

3.4.1 *Success rates – get_src*

The method calculates the cumulative frequency distribution of the the test cell susceptibilities (v_dist) for each fold. The results of the calculation for the folds are stored in the success_rates array and the src_df pandas DataFrame as a table. The curves of to these cumulative frequency distributions are referred here as the success rate curves (Figure 4).

3.4.2 *Area under the curve – get_auc*

The quality of the results can be evaluated by measuring the area under the success rate curves. The get_auc calculates the AUC value for the folds, along with the mean AUC of the folds and the standard deviation. This metric can be used to compare different versions of the analysis setup to each other in order to find the set of variables and parameters that give the most accurate results.

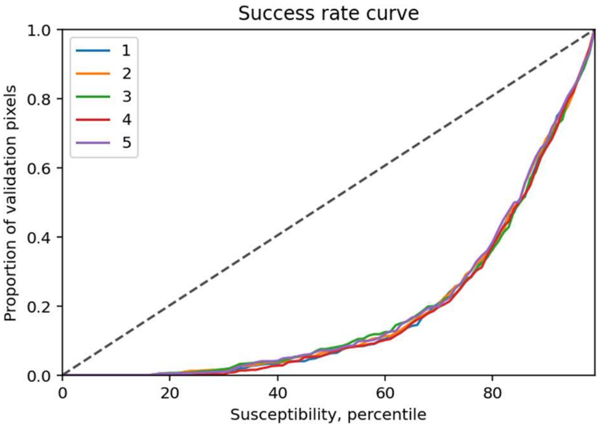

Figure 4. Cumulative frequency distribution of the test cells in the susceptibility categories. The “success rate curve”.

3.5 Output

The frmod script produces and stores several different grids and statistics during its run. These can be handled inside the Python environment or exported and used outside of it with external tools. Some basic utilities for plotting are also built-in, so the users can visualize the results without external tools.

3.5.1 *Grid display and export*

The main results of the analysis are the weighted grids and the susceptibility grids computed from them. The show_grid function of frmod's analysis module plots these grids (2D arrays with similar shape to the input rasters) as Matplotlib plots. Examples for these plots are on Figure 2 and Figure 3.

The export of the grid arrays to raster files is done by the utils module's array2raster function. The array2raster takes four input parameters.

- rasterfn: Path to a raster file. The new raster inherits its grid system and projection.
- new_raster_fn: Path to the new raster.
- array: The array to export.
- driver: The name of a GDAL raster driver. E.g., 'SAGA', 'GTiff'. The driver and the file extension should match.

3.5.2 *Display, analysis, and export of the statistics*

The supplementary statistics are also stored along the grid arrays, including the detailed frequency (or likelihood) ratio statistics and the evaluation metrics for each fold. These are stored as NumPy arrays or pandas DataFrames inside the FRAnalysis object. Both can be saved with their built-in export methods. The FRAnalysis also has functions for saving some of the statistics. The save_stats exports the weights computed for the analyzed variables to csv files. The save_src saves the success rates for each fold to one csv file.

The plot_var_fold_fr method plots the frequency distributions and their ratio on a combined figure (Figure 5). The input parameters for the plot_var_fold_fr function:

- name: The name of the chosen VRaster.
- fold: The fold ID, starting from 0.

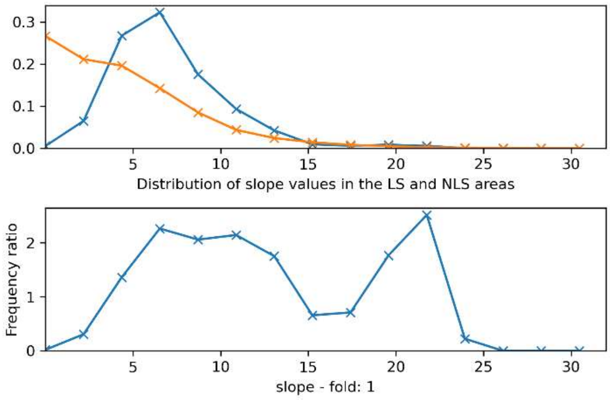

Figure 5. The frequency distributions and their ratio plotted with the FRAnalysis.plot_var_fold_fr method.

4. Distribution and installation

The code of frmod is hosted on Github at https://github.com/gerzsd/frmod. The freely available code is distributed under the MIT license. The Github repository also stores the documentation, a sample script, a set of sample input files, and an interactive user guide. The user guide is a Jupyter notebook (Kluyver et al., 2016), an interactive document where the executable code is accompanied by explanatory text and visualizations. The notebook can be run locally with the Jupyterlab application or online with Binder. Running the notebook online requires no setup from the user side, which makes testing the capabilities of frmod relatively quick and easy, even for researchers not familiar with Python programming. The online version of the interactive user guide is available from:

https://mybinder.org/v2/gh/gerzsd/frmod/main?filepath=frmod_demo.ipynb

The frmod package can also be installed locally from the distribution archives in the code release on Github. The code works with Python 3.7 and it is dependent on the following Python packages: NumPy, pandas, Matplotlib, GDAL. It is recommended to set up a separate environment for frmod to avoid conflicts with other packages.

5. Conclusions

frmod is a Python package developed for statistical landslide susceptibility assessment with methods established in the literature – the frequency and likelihood ratios. The input data for the script are the landslides of the study area and the analyzed continuous or thematic variables, both as raster grids. The statistical methods used for estimating landslide susceptibility mark those areas the most susceptible to landslides that are the most similar to the sample landslides based on the analyzed variables. The output of the script is a susceptibility grid (array) that can be used for creating a landslide susceptibility map. The final output is supplemented with other grids and statistics computed during the analysis. The analysis is run with cross-validation, and the results are evaluated based on the distribution of the test landslide cells in the susceptibility categories.

The goal of creating frmod was to make a tool that can carry out the whole workflow of the analysis from reading the input data, through estimating susceptibility, to evaluating the results, and producing supplementary statistics. Following through this workflow with desktop GIS softwares is usually relatively time-consuming and often requires exchange between multiple applications, which makes the process prone to user errors. The procedure becomes especially lengthy if multiple analysis setups are tested, which might make researchers to stick with the initial parameters instead of experimenting with variations that might produce better results. The motivation for developing this tool was to save work and time for its users by streamlining and automating the analysis workflow.

frmod achieves this goal as a Python script. However, the price of the smoother workflow is the lack of the familiar "clickable" user interface of the desktop GIS applications. This possibly limits the reachable user base because the package is harder to use for researchers not familiar with programming. On the other hand, running the analysis as a script offers more customizability and room for experimenting. The results of the analysis can be further explored with the growing palette of Python data science tools (e.g., pandas), while the outputs (text files and raster grids) remain compatible with the more common GIS tools. The interactive user guide provides example code for setting up a susceptibility analysis with frmod. As the guide can be run online, interested users can try out the script without installing it locally. Of course, a local installation is required for using frmod with own data.

The script provides evaluation metrics during its run to test the predictive performance of the results. These metrics allow to compare the different iterations of an analysis to each other. One aim for frmod's further development would be to include the testing of the usefulness and importance of the different input data and parameters on the results. It is also important to point out that while frmod tests the performance of the analysis, the careful preparation of input data and thorough field checking of the results must not be overlooked during the susceptibility assessment.

6. References

Brabb, E. E. (1984). Innovative Approaches to Landslide Hazard Mapping. *Proc. 4th Int. Symp. Landslides, Toronto*, 307–324.

Chung, C. (2006). Using Likelihood Ratio Functions for Modeling the Conditional Probability of Occurrence of Future Landslides for Risk Assessment. *Computers & Geosciences, 32*(8), 1052–1068, https://doi.org/10.1016/j.cageo.2006.02.003

Erickson J., Daniel, C., and Payne, M. (2013). Python GDAL/OGR Cookbook 1.0 Documentation. Retrieved 16 March 2021, from http://pcjericks.github.io/py-gdalogr-cookbook/

GDAL/OGR contributors (2021). GDAL/OGR Geospatial Data Abstraction Software Library. Open Source Geospatial Foundation. https://gdal.org

Guzzetti, F., Carrara, A., Cardinali, M., and Reichenbach, P. (1999). Landslide Hazard Evaluation: A Review of Current Techniques and Their Application in a Multi-scale Study, Central Italy. *Geomorphology, 31*(1–4), 181–216. https://doi.org/10.1016/S0169-555X(99)00078-1

Harris, C. R., Millman, K. J., van der Walt, S. J., Gommers, R., Virtanen, P., Cournapeau, D., Wieser, E., Taylor, J., Berg, S., Smith, N. J., Kern, R., Picus, M., Hoyer, S., van Kerkwijk, M. H., Brett, M., Haldane, A., del Río, J. F., Wiebe, M., Peterson, P., … and Oliphant, T. E. (2020). Array Programming with NumPy. *Nature, 585*(7825), 357–362. https://doi.org/10.1038/s41586-020-2649-2

Hunter, J. D. (2007). Matplotlib: A 2D Graphics Environment. *Computing in Science & Engineering, 9*(3), 90–95. https://doi.org/10.1109/MCSE.2007.55

Kluyver, T., Ragan-Kelley, B., Pérez, F., Granger, B., Bussonnier, M., Frederic, J., Kelley, K., Hamrick, J., Grout, J., Corlay, S., Ivanov, P., Avila, D., Abdalla, S., Willing, C., and Jupyter development team. (2016). Jupyter Notebooks – A publishing format for reproducible computational workflows. In F. Loizides and B. Scmidt (Eds.), *Positioning and Power in Academic Publishing: Players, Agents and Agendas* (pp. 87–90). IOS Press. https://www.doi.org/10.3233/978-1-61499-649-1-87

Lee, S. and Talib, J. A. (2005). Probabilistic Landslide Susceptibility and Factor Effect Analysis. *Environmental Geology 47*(7): 982–990. https://doi.org/10.1007/s00254-005-1228-z

McKinney, W. (2010). Data structures for statistical computing in python. *Proceedings of the 9th Python in Science Conference*, 56–61. https://conference.scipy.org/proceedings/scipy2010/pdfs/mckinney.pdf

Parise, M. (2001). Landslide Mapping Techniques and Their Use in the Assessment of the Landslide Hazard. *Physics and Chemistry of the Earth – Part C: Solar Terrestrial & Planetary Science, 26*(9), 697–703. https://doi.org/10.1016/S1464-1917(01)00069-1

Reichenbach, P., Rossi, M., Malamud, B. D., Mihir, M., and Guzzetti, F. (2018). A review of statistically-based landslide susceptibility models. *Earth-Science Reviews, 180*, 60–91. https://doi.org/10.1016/j.earscirev.2018.03.001

Soeters, R. and van Westen, C. J. (1996). Slope instability recognition, analysis, and zonation. In A. K. Turner and R. L. Schuster (Eds.), *Landslides, investigation and mitigation* (Transportation Research Board, National Research Council, Special Report; 247) (pp. 129-177). National Academy Press.

Thiery, Y., Terrier, M., Colas, B., Fressard, M., Maquaire, O., Grandjean, G., and Gourdier, S. (2020). Improvement of landslide hazard assessments for regulatory zoning in France: STATE–OF–THE-ART perspectives and considerations. *International Journal of Disaster Risk Reduction, 47*, Article 101562. https://doi.org/10.1016/j.ijdrr.2020.101562

Varnes, D. J. (1984). Landslide Hazard Zonation: A Review of Principles and Practice. UNESCO, Paris.