Identification of similar catchments (HADES)

Creating a tool for interactive selection of similar catchments in Python

Seminar “Geodata Analysis and Modelling”
Institute of Geography
University Bern
FS 2019

Supervisors:

PD Dr. Andreas Paul Zischg
andreas.zischg@giub.unibe.ch

Dr. Pascal Horton
pascal.horton@giub.unibe.ch

Students:

Flavia Polli
15-115-355
flavia.polli@students.unibe.ch

Eileen Schilliger
15-104-516
eileen.schilliger@students.unibe.ch

Thursday, 21st November 2019

Project overview

This report was created as part of the course geodata analysis and modeling and should be seen as a first exercise to gain experience in programming in Python. The aim was to write a script, which should enhance the user experience on the web platform HADES (the Hydrological Atlas of Switzerland). HADES contains a range of hydrological information of Switzerland, including didactic materials, excursion guides, and interactive maps. The platform is accessible for free and is the result of a collaborative effort of Swiss hydrologists managed by the Institute of Geography at the University of Bern.

On the interactive map, a catchment can be selected by the user. As a result, the properties of the selected catchment are displayed. Additionally, the goal is to be able to present similar catchments regarding the chosen catchment ID. Therefore, a script was developed that detects similar catchments based on freely selectable catchment characteristics.

Scientific setting

The scientific value of this project was not the focus of our work, as it covers a more application-oriented topic. It is an exercise which is not related to a master thesis or any other research project. Hence, there isn't scientific literature that this project could be embedded into. However, there certainly is a value in this project, as it contributes to the goal of providing a broad audience with scientific data. Data that is presented in an appealing and readable fashion is more likely to be seen by a wider audience.

Input data

As the used Dataset has open access, it could be downloaded directly from the HADES website. The shapefile data is available in multiple files categorized in the area size of the catchments. The work was deliberately limited to one catchment area. We chose to work with the dataset that includes all catchments with an area of approximately 200 m^2, as we figured that it includes enough catchments that one could be compared too. On the other hand, the dataset is still small enough to avoid unnecessary enlargement of computing time. With the program *QGIS* the downloaded dataset could be opened and the attribute table was extracted as a text file (csv). This csv Dataset was later imported into the Python script. The only problem we faced with the file was, that the headers aren't clearly labeled. We had to make assumptions on which column contains which information. Therefore, the receiver of the script (that has detailed knowledge of the datasets from HADES) has to go through our proposed list of column header assumptions and check and complete it. The python script can be transferred to the other shapefiles.

Approach and method

For this script we used the program PyCharm CE (Version 2018.3.5), which was introduced in the seminar *Geodata Analysis and Modelling*. In Python we decided to work with Pandas because this library is well suited for data analysis (GeeksforGeeks, 2019).

Because we never programmed anything before, we used tools like the internet or books as sources of information. The main procedure was to start only with one variable. Only when this variable was successfully programmed other variables were added. This made it possible to test the functions one by one and to in the end run them in loops.

There are multiple options on how to detect similar catchments based on freely selectable catchment characteristics. After a meeting with the supervisor of the seminar it was decided to work with quantiles to be able to group the different catchments regarding their similarity. Therefore, no limit values had to be determined manually and the user can choose how many quantiles he wants.

Challenges

The first challenge already occurred while reading-in the file path of the datafile into PyCharm. The excel file used at first contained empty fields and commas, which caused error messages. Thus the file was converted to a csv format (“Macintosh-kommagetrennt csv.”) to enable the datafile to be imported with the function `pd.read_csv`.

There were multiple problems with various functions. especially the choosing and using functions caused difficulties, because we have never programmed before. For example, we did not know exactly by which function a selection of several criteria could be programmed in a data matrix. The support of our supervisors and to use the tips on public python platforms in the internet helped us to figure out the right functions and to write the concluding script.

Description

After importing the dataset and packages **pandas** and **numpy**, the next few lines of code enable the user to select a catchment and the number of quantiles the script should work with. In a further step a list with catchment characteristics – resembling middle hight, slope or aspect – is presented to the user (called *display_liste*). The user can select one or multiple characteristics he wants to focus on from that list. For the ongoing calculation, the script doesn't use the presented list. It is only there for display purposes, as the headers in the data file that the script works with, are not self-explanatory and therefore can't be presented to the user to choose from. It is crucial for the success of the calculation that both lists (*display_liste* and *db_list*) are referring to the same headers/characteristics of the data file in the same order.

```
display_liste =  ['middle Hight', 'slope',         'aspect'…        ]
db_list =        ['mH_04',        'N20slp8_12',    'asp8smm_14'…    ]
```

The receiver of the script (that has detailed knowledge of the datasets from HADES) has to go through both proposed lists and check if, for instance, it is accurate that the variable `'mH_04'` represents the middle hight. The receiver of the script has to correct and add additional variables to both lists before use.

The quantiles of the selected variables are calculated with the function **qcut** and saved into additional columns in the dataset. The script now goes through each catchment in the dataset and detects, if the calculated quantiles of the variables are equal to the ones from the selected catchment. If this is the case the processed catchment-id is added to the final list (*ergebnisliste*) as it is seen as similar to the catchment chosen by the user.

In the end, the catchment-ID the user selected in the beginning of the script, has to be removed from the *ergebnisliste* with the function **remove**. Then the final list of all similar catchments is presented to the user.

Short Interpretation

First we created the script for the 200m^2 dataset and then we extended it to other catchment areas (100m^2, 150m^2, 300m^2). A brief analysis of the different catchments was carried out (see annex to this work). The aim of this study was to find out to what size of catchment area and number of quantiles the script could be used. For datasets that include catchments with an area of 200m2 or smaller, the script can present similar catchments. However, the datasets are too small for the user to be able to choose too many variables simultaneously or higher quantiles. With an input of 5 quantiles the script still works well, but 8 quantiles are clearly too high. With too many conditions no similar catchments can be detected. We, therefore, recommend to only use the script with datasets that include equal or more catchments than the dataset with catchments of 200m^2 (the smaller the catch–ments, the more catchments are included in the dataset). The selection criteria *slope* often does not identify common catchment areas. This may be because for this aspect are not many deviating values. Are fewer catchments in a dataset, the similarity between the chosen catchment-id and the output-list would be very weak, as you would need to run it with for example quantiles as low as 2. If there are no catchment areas with the same conditions for the selected id, no result will be displayed by the script. If, however, the individual catchment areas have equal conditions (slope, aspect and middle hight), but the combination of several conditions does not display the same areas, the result 0 will be displayed.

Further, it is worth noting again, that this approach is purely statistical. No expert knowledge of the hydrological topic is integrated, which would have certainly increased the quality of the result. However, relying on quantiles presented itself to be a practical solution as other statistical methods or incorporation of expert knowledge would have been much more labor-intensive. Which wouldn’t have been commensurate, as this project is only used for display purposes on a website. As a compromise, we suggest that an expert from HADES has a look at the output-lists and determines if the result does fulfill his or her requests.

As this project is our very first time to program with python, the script could certainly be improved and shortened by advanced python user. But as the data files aren’t too big and the script will not run in the background of the HADES website it shouldn’t matter that the script includes a few loops.

Outlook

The receiver of the script (that has detailed knowledge of the datasets from HADES) has to go through the script, make the changes to the highlighted lists and run it with desired quantiles and variables. After a check on the similarity of the catchments by the user, the output list can be embedded on the webpage HADES to enable the additional display function.

The script in general can possibly be transferred to other topics, for which automatic selection is to be made.

Sources

GeeksforGeeks (2019). Python | Data analysis using Pandas. https://www.geeksforgeeks.org/python-data-analysis-using-pandas/ (01.10.2019).

Appendix

Script tests with different catchment sizes (datasets).

Dataset with 100m^2-area catchments

Feature ID for testing the script: 144’174

100m^2, 3 Quantiles

Middle Hight	X			X	X		X
Slope		X		X		X	X
Aspect			X		X	X	X
Number of similar Catchments	48	46	49	38	18	17	14

100m^2, 5 Quantiles

Middle Hight	X			X	X		X
Slope		X		X		X	X
Aspect			X		X	X	X
Number of similar Catchments	28	27	28	7	5	5	3

100m^2, 8 Quantiles

Middle Hight	X			X	X		X
Slope		X		X		X	X
Aspect			X		X	X	X
Number of similar Catchments	17	-	18	-	0	-	-

Dataset with 150m2-area catchments

Feature ID for testing the script: 111’680

150m^2, 3 Quantiles

Middle Hight	X			X	X		X
Slope		X		X		X	X
Aspect			X		X	X	X
Number of similar Catchments	32	32	32	17	11	6	4

150m^2, 5 Quantiles

Middle Hight	X			X	X		X
Slope		X		X		X	X
Aspect			X		X	X	X
Number of similar Catchments	19	32	18	8	5	3	1

150m^2, 8 Quantiles

Middle Hight	X			X	X		X
Slope		X		X		X	X
Aspect			X		X	X	X
Number of similar Catchments	12	-	11	-	2	-	-

Dataset with 200m2-area catchments (main script)

Feature ID for testing the script: 154’921

200m^2, 3 Quantiles

Middle Hight	X			X	X		X
Slope		X		X		X	X
Aspect			X		X	X	X
Number of similar Catchments	24	19	8	13	25	34	10

200m^2, 5 Quantiles

Middle Hight	X			X	X		X
Slope		X		X		X	X
Aspect			X		X	X	X
Number of similar Catchments	14	4	1	4	15	16	3

200m^2, 8 Quantiles

Middle Hight	X			X	X		X
Slope		X		X		X	X
Aspect			X		X	X	X
Number of similar Catchments	9	-	-	-	9	-	1

Dataset with 300m2-area catchments

Feature ID for testing the script: 179’387

300m^2, 3 Quantiles

Middle Hight	X			X	X		X
Slope		X		X		X	X
Aspect			X		X	X	X
Number of similar Catchments	15	22	15	13	4	10	4

300m^2, 5 Quantiles

Middle Hight	X			X	X		X
Slope		X		X		X	X
Aspect			X		X	X	X
Number of similar Catchments	9	-	8	-	1	-	-

300m^2, 8 Quantiles

Middle Hight	X			X	X		X
Slope		X		X		X	X
Aspect			X		X	X	X
Number of similar Catchments	5	-	5	-	1	-	-