

Jupyter Technology as a Cartographer's Work Environment: A Case Study

Kamil Choromański^{a,*}, Dariusz Gotlib^b, Jakub Łobodecki^c

a Warsaw University of Technology, Faculty of Geodesy and Cartography, kamil.choromanski@pw.edu.pl,

b Warsaw University of Technology, Faculty of Geodesy and Cartography, dariusz.gotlib@pw.edu.pl,

c Warsaw University of Technology, Faculty of Geodesy and Cartography, jakub.lobodecki@pw.edu.pl,

* Corresponding author

Abstract: The article describes research experiments carried out to test the feasibility of using Jupyter technology as a tool to assist cartographers. The case study includes several examples typical of the stage of data preparation and analysis prior to the execution of the cartographic presentation itself. At the same time, the possibility of performing simple cartographic presentations useful at the stage of analysis performed by various researchers is presented in this environment. The purpose of the cases discussed is not to perform a complex and typical cartographic task but to highlight Jupyter Notebook's usefulness in cartographers' work.

The authors introduce the essential features of this technology, briefly describe the experiments performed using geospatial cyberinfrastructure, and demonstrate the overall potential and limitations of the Jupyter environment. Its most important advantages include a possibility of prototyping as part of the map development process, automation of geodata preparation processes, and facilitation of teamwork and sharing work results. The experiments confirmed the usefulness of the technology, although they also demonstrated some particular requirements and barriers in its use. Subsequent research teams can use the Jupyter notebooks that were created as part of the experiment.

Keywords: Jupyter notebook, cartography, interactive maps, spatial data science, geoinformation

1. Introduction

The cartographer's work environment and tools are constantly changing. Development of new cartographic products requires use of different technologies. Needs and opportunities also change over time. Looking at the last few decades, we see a shift in emphasis from technologies that allow professional printing of maps on paper, to digital solutions based on graphics software, and then to technologies that rely on data geovisualization from a database. In the last case, we are also witnessing a clear shift from applications based on desktop GIS software to cloud-based applications. Cartographic studies are increasingly based on large, diverse, and rapidly changing data, and often on what is referred to as big data. The analyses that are carried out before preparation of a cartographic presentation are even more complex. Another important innovation is increasing interactivity of maps and use of various information media. Working on a virtual computer with access to vast data resources at one's fingertips is becoming increasingly common. Newer and more sophisticated tools are now available that can support preparation of cartographic studies with the above characteristics. Jupyter technology, for example, is such a tool, that powers products such as Google's Colab, Jupyter Notebook, and JupyterLab. It is increasingly being used by data science specialists.

The term "Spatial Data Science," which is a combination of Spatial Science and Data Science, has already been established as well (Wilkening, 2019). Wilkening also pointed out that "cartography and GIS tools are the necessary technological building blocks for solving the (80% spatial) data science problems of the future."

Still, Jupyter technology is not widespread or well-known in the cartographic community. One of the examples may be lack of mention of this technology in a broad review of *Multimedia Cartography* articles (Medynska-Gulij et al., 2022).

However, it is a technology mature enough to interest the cartographic community. A dedicated map widget has already been built based on the popular Leaflet¹ mapping library which is an interactive map library originally written in JavaScript for use in desktop and mobile web applications. Its version for Jupyter is called "ipyleaflet" and is officially supported by Jupyter (GitHub Jupyter-Widgets/Ipyleaflet, 2014/2022).

Jupyter has also been successfully used in spatial data analysis tasks, as well as data visualization (Marchi et al., 2017). Examples of data collection, import, processing, and visualization processes based on this environment are available. A good illustration is using a sizeable data-sharing service based on ISO standards to conveniently search and explore data (Valentine et al., 2021).

¹ <https://leafletjs.com>

The Leafmap library is worth noting; it is an open-source Python package that enables users to analyze and visualize geospatial data with minimal coding in a Jupyter environment (Wu, 2021).

Few publications indicate that Jupyter technology was used in preparation of visualizations. An interesting example of where Jupyter (with use of the Google Colab² service) was used is the preparation of a collaborative emotional map by Camara et al., (2021). In this project, the ready-to-use Notebook was published on GitHub – which presents another advantage of this environment: the ease with which analysis can be shared.

Jupyter is also mentioned as one of the tools enabling research into combining Earth Science with Spatial Big Data (Madhukar & Pooja, 2019).

Two interesting uses of map technology in the Jupyter environment are the interactive visualization of sky maps (Weixiang et al., 2017) and the interactive visualization of graph network dependencies (Rosenthal et al., 2018).

One more indication that this technology is little used in the cartographic community is the dual occurrence of the keywords “jupyter” and “cartography” in the statistics of scientific publications. Figure 1 shows a chart of publications containing the keyword “jupyter” from the Scopus database of published scientific papers. When combined with the keyword “cartography” the number of publications drops to 1.

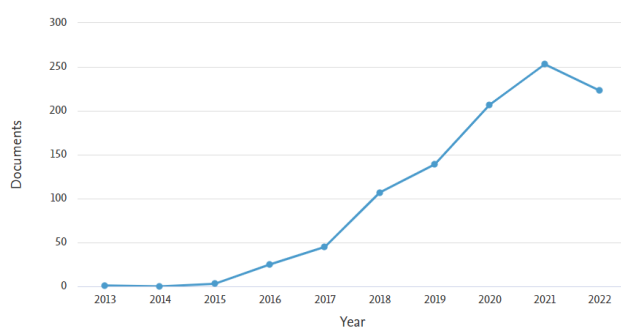


Figure 1. The number of publications in the Scopus database for the query “jupyter” per year. Source: scopus.com.

The number of publications is only slightly higher when combining keyword “cartography” with keyword “python” (Figure 2), which may indicate general lack of research interest in this area.

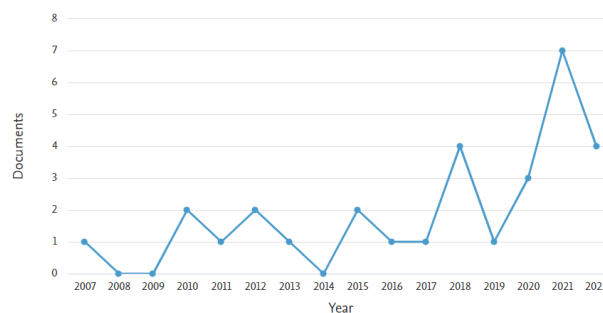


Figure 2. The number of publications in the Scopus database for the query “python AND cartography” per year. Source: scopus.com.

This article aims, therefore, to highlight and demonstrate selected opportunities for utilizing this technology in cartographic work by providing examples of its applications.

Of course, the question of the essence of cartographic work is still debatable. Sometimes it is limited only to cartographic presentation in a graphic form. However, the authors are convinced that the boundary between work of a cartographer, a GIS expert, or a geoinformation analyst is becoming increasingly blurred. Almost all cartographic products are created based on databases that often meet the condition of being so-called big data and require complex spatial analysis.

Due to the limitations of the volume of this article, the examples given hereafter will be simple to illustrate the need and possibilities of using technologies such as Jupyter Notebook.

2. Jupyter technology in the cartographic context

Jupyter technology emerged relatively recently, i.e., around 2014³, evolving from the *IPython* project, which focused on creating an interactive Python language shell for rapid testing and prototyping. The Jupyter project has expanded the original premise to include a web-based interactive development environment that combines the ability to execute code and to store text, images, data, and other multimedia. It is a community project that aims to develop free software, open formats, and services for interactive computing.

The primary file used in the Jupyter environment is called the “Jupyter Notebook” (.ipynb extension). Using a single file, the user can write code in various programming languages (Python by default), display data and calculation results, and write text with explanations about, for example, ongoing calculations. The Jupyter Notebook is composed of cells. The type of a cell defines its content: the two main cell types are “Code” and “Markdown.” In cells of the Code type, it is possible to write and execute code. The result of the code’s execution will appear under the cell. Markdown type cells allow the code to be supplemented with a descriptive part in the form of a

² <https://colab.research.google.com>

³ <https://jupyter.org/about>

markup language of the same name. This language allows insertion of such elements as formatted text, images, tables, or LaTeX elements (e.g. formulas). Internally, the Jupyter Notebook document is stored in JSON format and is easily transferable between computers and shared over networks.

The Jupyter environment comes in two versions: Jupyter Notebook and JupyterLab. The former is characterized by a more modest, less elaborate interface. The latter, has a stronger resemblance to the classic development environment in its design. However, both versions use the same core software solution (*IPython*) and have identical functionality.

Jupyter is an environment that can be utilized by different industries. The specifics of this environment make it a tool that is widely used by data scientists. J. Wilkening (2019) points out that “to bridge the gap between spatial sciences (such as GIS and cartography) and data science, we need an integrated framework of spatial data science.” And, in this context, he points to Jupyter as being one of the essential tools in this process.

From the cartographic perspective, this tool allows the calculations that precede cartographic visualizations in an interactive mode to be performed. The parameters can be adjusted by changing the values entered in the Notebook. Complex data processing can also be performed. The appropriate use of Jupyter and programming libraries that allow data visualization, makes it possible to observe the effect of various data parameters on the resulting image. Therefore, it is an excellent tool for prototyping and experimentation; so that later, the complete target

calculations can be performed in a different environment that is optimal for a specific application.

However, unlike classic GIS tools, it requires knowledge of programming basics from the very beginning, even to perform the simplest operations. Nevertheless, using languages such as Python lowers the entry level significantly and allows even those who are inexperienced in programming to obtain results quickly.

Jupyter Notebook is also a tool for describing and documenting research, and sharing ideas and algorithms, as well as simultaneously verifying research and experiments through reproducibility by subsequent project or research teams.

The list of cartographic tasks that can be supported through the use of Jupyter technology seems to be broad.

Most typical cartographic work begins with data being selected from multiple sources, checking their quality, harmonizing the data in the context of a specific cartographic product, analyzing the data, detecting trends or processes, generalizing the data, and finally, formulating the proper cartographic message, including various types of geovisualization.

At the present time, cartographic work usually requires extensive data processing. An example of this would be performing calculations for a spatial accessibility map of an entire country or continent. The final generated cartographic image must be preceded by calculating millions of potential travel routes (Figure 3).

Another example would be use of artificial intelligence methods for identifying objects, comparing different

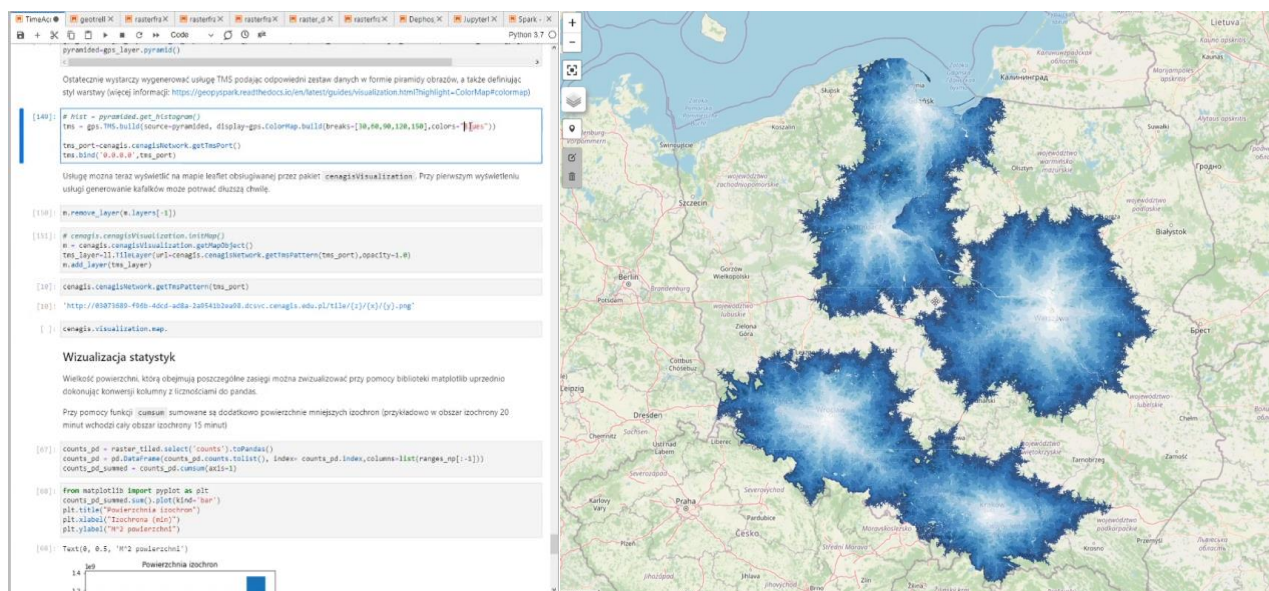


Figure 3. An example of using Jupyter Lab for the calculations needed to generate a spatial accessibility map (on the left is the code in Python and a description of the procedures being performed, and on the right is a window with the result).

datasets, and detecting anomalies in those datasets (e.g., forgeries), through to the automated, optimal generation of descriptions related to the presented objects.

Choosing a visualization method also often requires performing several trials – sometimes dozens or hundreds.

In this context, Jupyter is worth considering since it was created exactly for this purpose.

Jupyter can also be an excellent educational tool. A student or trainee can simultaneously read the manual and, in the same environment, run software and programming scripts and then view the results. The use of the Jupyter environment in educational activities is documented in the literature, for example, in *Weather Forecasting Class* (Galen et al., 2022).

3. Case study: Experiments in spatial data combined with geovisualization

In order to test the capabilities of the Jupyter technology, a decision was made to perform a series of experiments and tests corresponding to frequent GIS/cartographic analytical tasks. The most important of these include creating a map of building heights, visualizing the area of administrative units in the form of a cartogram, supervised learning based on a manually mapped learning set, visualizing Floating Car Data (FCD) in the form of Level Of Service index values, generating a Topographic Position Index (TPI) raster based on a digital elevation model, and developing a temporal range maps.

Due to the limitations of the volume of this article, only the results of the first three simple experiments are presented below. Figures 4, 5, 6 and 7 are intended to show the general appearance of the Jupyter's geospatial analysis environment.

3.1 Selecting the working environment

In order to carry out the study, a decision was made to use the experimental CENAGIS Platform. This platform can be described as a geospatial data cyberinfrastructure. The platform forms the basis for the operation of the Centre for Geospatial and Satellite Analysis. This innovative cloud-based technology solution, implemented at Warsaw University of Technology, provides four main subsystems: a virtualization subsystem, a spatial big data subsystem, a geo-applications subsystem (geoinformation hub), and a geospatial data repository. The platform is based on technologies such as CloudStack, Hadoop Distributed File System (HDFS), Apache Spark, and JupyterLab. At the heart of the platform are two clusters named after cartographers: MERCATOR and ROMER. One of the advantages of this platform is a completely configured Jupyter environment, with access to multiple software libraries and rich vector, raster, and point cloud data resources for the entire Polish area, sourced from the National Spatial Data Infrastructure, community resources (e.g. OpenStreetMap), and the European Space Agency satellite data repository. The solution is partially similar to the existing CyberGIS platform in the USA, which has also

implemented support for Jupyter technology as one of its standard geospatial analytics tools (Yin et al., 2019).

3.2 Case 1: Administrative unit area visualization

The first and simplest use case involved calculating and visualizing, in the choropleth map form, the area of each municipality located in Poland. The data was taken from the Accumulo (Kepner et al, 2014) NoSQL database, and the calculation was performed in a distributed manner using Apache Spark and the GeoMesa (Hughes et al, 2015) package, based on vector data from the Polish National Register of Boundaries. The data was then transferred to a single computing machine and visualized using the *ipyleaflet* programming library. The result is shown in Figure 4. In addition to the data calculations, the Jupyter technology made it possible to conveniently and quickly change basic aspects of the visualization, such as, the colour palette, adding a legend, and dynamic visualization based on attribute changes.

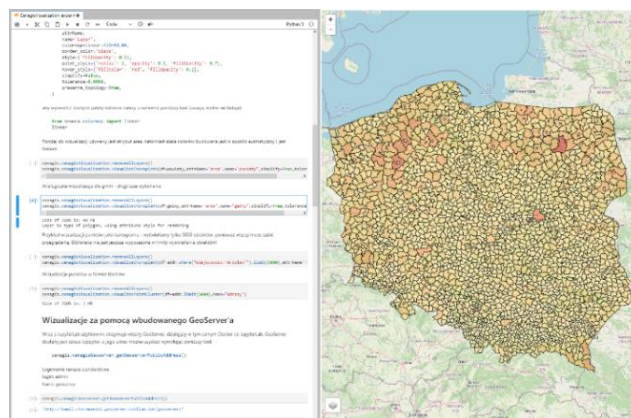


Figure 4. The choropleth map of administrative unit areas created using JupyterLab and *ipyleaflet*.

The experiment showed that in a properly configured Jupyter environment, big spatial datasets can be quite easily used, quick calculations can be made, and they can be immediately visualized using cartographic methods.

3.3 Case 2: Developing maps based on DEM and DSM model

In this case, multiple data sources were used:

- a) vector building location data from the Polish Database of Topographic Objects (BDOT10k);
- b) a Digital Elevation Model (DEM) in the raster form, from the resources of the Head Office of Geodesy and Cartography;
- c) a Digital Surface Model (DSM) in the form of a raster, from the resources of the Head Office of Geodesy and Cartography.

The analysis was conducted for the area of Poland's capital – Warsaw. As in the case of the first analysis, the capabilities of Apache Spark (Zaharia et al, 2010) software were used to split the calculations over multiple computational units. Using the RasterFrames package, both raster layers were loaded, and then the DEM was subtracted from the DSM to obtain a normalized DSM,

which represented the building heights. Then a spatial concatenation of the vector data about the buildings and the obtained raster was performed, which transferred information about the average raster value in the area of each building.

In order to efficiently visualize such a large number of objects on a map, GeoServer software was used to display the result as a WMS layer. The function for dynamically creating a WMS service from data analyzed using Jupyter technology was created as part of the work on the Spatial Big Data subsystem. A section of the developed notebook, with code and visualization, is shown in Figure 5.

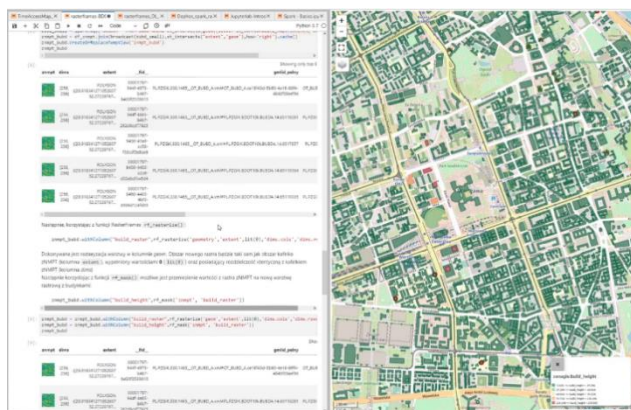


Figure 5. Visualization of building height analysis using a WMS layer, created based on the Jupyter analysis' results.

The experiment showed that in the Jupyter environment, it is possible to effectively analyze big datasets of DEMs and DSMs to develop various cartographic presentations on this basis. The map of building heights is just an example of using the discussed environment.

3.4 Case 3: Terrain classification based on a machine learning method

The third use case focused on an analysis using images from the Sentinel-2 satellite, obtained from the ESA satellite data repository. This is an example of using the interactive capabilities of the *ipyleaflet* module for land/sea segmentation.

The ability to manually draw shapes on the map was used, in this case, to collect learning samples from the input satellite image (Figure 6). The user created two polygons – one representing a sample of data from the land, and the other a sample of data from the sea. This data was then fed into a random forest model (Breiman, 2001), which was used to classify all the other pixels in the image based on their RGB color values. The Scikit-learn library was used to implement the machine learning task. The final result was visualized on the same map (Figure 7).

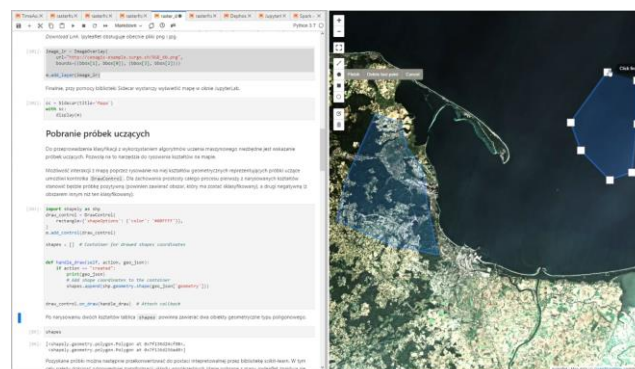


Figure 6. The process of creating training samples using an interactive map in JupyterLab.

The use of both interactive maps and Jupyter technology in this case allowed an easy introduction to machine learning technology, which is increasingly used in cartography, and makes it possible to do quick experiments on the relationship of the size and diversity of the learning data to the quality of the result.



Figure 7. Result of a random forest classification model trained using training samples drawn on an interactive map.

Data classification and identification of objects from satellite and aerial images is one of the important data preparation processes in creation of cartographic products. The ability to automate this process and parameterization using Jupyter can be very convenient and useful for cartographers.

3.5 Discussion of results

In all the described cases, the intended purpose was achieved. It was possible to load, combine, analyze, and visualize a diverse range of geospatial raster and vector data. The use of the Jupyter environment allowed rapid prototyping and refinement of the results, as well as the description of individual code fragments, which facilitated the use of the notebooks by others, making them valuable as an educational and research tool.

The basic setup of the Jupyter environment is relatively easy and can be done with a standard personal computer. However, in such a configuration this tool cannot achieve the results shown above and is not optimal from the perspective of cartographic research. Access to consistent and up-to-date spatial data resources from multiple sources and types may be necessary in order to accomplish a variety of tasks. Moreover, there is an increasing need to analyze big spatial data, which may require additional computing resources and special tools such as Apache

Spark. In some of the cases described, this technology was used to speed up calculations and scale them up. Finally, additional components, such as the *ipyleaflet* programming library or a GeoServer connection, are needed in order to visualize the results of work done in the Jupyter environment. It was possible to automate and simplify many cartographic analysis-relevant tasks, such as publishing WMS, TMS services, or reading large datasets from a NoSQL database.

The aforementioned aspects mean that, for a cartographer, setting up an optimal working environment using Jupyter technology is not a simple task, and use of research platforms such as mentioned above, where such environments are already configured, is much more efficient.

In addition to the difficulty of configuring such an environment, the study found other issues. The visualization capabilities introduced by *ipyleaflet* are sufficient for quick prototyping and the preliminary assessment of results, however, it was not designed for creating professional products capable of fulfilling the high standards of a cartographic edition. This task is still more easily achieved using standard GIS tools. Moreover, because Jupyter and *ipyleaflet* rely on web browsers for their operation, performance and memory problems are also possible, particularly when visualizing a very large number of objects on a map.

However, advantages such as the ability to immediately view results after changing input parameters, the ease of sharing the obtained results, the possibility of creating reproducible experiments in the form of Jupyter Notebooks, mean that this solution has great potential for facilitating and accelerating some of the work of cartographers.

Presented notebooks are available publicly on GitHub⁴, however, they are designed to work specifically inside CENAGIS spatial big data computing environment.

4. Conclusions

Jupyter technology is not widely known or used by cartographers, as an analysis of the literature shows. At the same time, Jupyter is a modern tool used increasingly by thousands of scientists and data scientists. The advantages of this technology, also for cartographic work, include the following:

- a) The ability to integrate the working environment with data scientists from outside the geoinformation industry and easily allow interdisciplinary cooperation;
- b) Facilitating access to big data resources and computing clusters;
- c) Ease of experimenting and prototyping through the ability to run parallel cartographic tasks in different

variants, to have automated supervision of task execution, and to allow team discussion of the results.

One of the factors that confirms the importance of this technology for the GIS/cartography user community is evidenced by the example of the inclusion of Jupyter technology in the ESRI ecosystem.

It is also important to keep in mind the barriers associated with using this technology. Using these tools requires cartographers to have a more profound knowledge of geoinformatics.

Cartography cannot remain indifferent to the development of such new technologies. It is necessary to include the issues raised in this paper in academic courses and training programs. It is also worthwhile making greater attempts to use this tool in cartographic research projects.

5. Acknowledgements

The authors are thankful for the opportunity to use the CENAGIS geospatial data cyberinfrastructure, developed under the project entitled "CENAGIS - The Center for Scientific Geospatial Analyses and Satellite Computations including Laboratories for testing and authorising geomatic products," which is co-financed from the European Fund for Regional Development within Priority Axis I "Utilization of research-and-development activities in economy" (RPMA.01.01.00-14-9871/17 from 11.12.2018).

6. References

- Breiman, L. (2001). Random Forests, Machine Learning volume 45, 5–32, <https://doi.org/10.1023/A:1010933404324>
- Camara, G. S., Camboim, S. P., & Bravo, J. V. M. (2021). Using Jupyter Notebooks for viewing and analysing geospatial data: Two examples for emotional maps and education data. The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences, XLVI-4/W2-2021, 17–24. <https://doi.org/10.5194/isprs-archives-XLVI-4-W2-2021-17-2021>
- Galen, L. van, Hartogensis, O., Benedict, I., & Steeneveld, G.-J. (2022). Teaching a Weather Forecasting Class in the 2020s. Bulletin of the American Meteorological Society, 103(2), E248–E265. <https://doi.org/10.1175/BAMS-D-20-0107.1>
- Hughes, James & Annex, Andrew & Eichelberger, Chris & Fox, Anthony & Hulbert, Andrew & Ronquest, Michael. (2015). GeoMesa: a distributed architecture for spatio-temporal fusion. 94730F. <https://doi.org/10.1117/12.2177233>.
- Kepner, J., Arcand, W., Bestor, D., Bergeron, B., Byun, C., Gadepally, V., Hubbel, M., Michaleas, P., Mullen, J., Prout, A., Reuther, A., Rosa, A. & Yee, C. (2014). Achieving 100,000,000 database inserts per second using

⁴https://github.com/KamilChoromanski/CENAGIS_Notebooks

- Accumulo and D4M. IEEE High Performance Extreme Computing Conference, <https://doi.org/10.1109/HPEC.2014.7040945>
- Github [jupyter-widgets/leaflet](https://github.com/jupyter-widgets/leaflet). (2022). [JavaScript]. Jupyter Widgets. <https://github.com/jupyter-widgets/leaflet> (Original work published 2014)
- Madhukar, M., & Pooja. (2019). Earth Science [Big] Data Analytics. *Big Data for Remote Sensing: Visualization, Analysis and Interpretation*, 99–128. https://doi.org/10.1007/978-3-319-89923-7_4
- De Marchi, D., Burger, A., Kempeneers, P., & Soille, P. (2017). Interactive visualisation and analysis of geospatial data with Jupyter. *Proc. of the BiDS*, 17, 71-74.
- Medynska-Gulij, B., Forrest, D., & Cybulski, P. (2022). *Multimedia Cartography*. International Journal of Geo-Information MDPI, Basel. <https://doi.org/10.3390/books978-3-0365-3679-8>
- Rosenthal, S. B., Len, J., Webster, M., Gary, A., Birmingham, A., & Fisch, K. M. (2018). Interactive network visualization in Jupyter notebooks: VisJS2jupyter. *Bioinformatics*, 34(1), 126–128. <https://doi.org/10.1093/bioinformatics/btx581>
- Valentine, D., Zaslavsky, I., Richard, S., Meier, O., Hudman, G., Peucker-Ehrenbrink, B., & Stocks, K. (2021). EarthCube Data Discovery Studio: A gateway into geoscience data discovery and exploration with Jupyter notebooks. *Concurrency and Computation: Practice and Experience*, 33(19). <https://doi.org/10.1002/cpe.6086>
- Weixiang, Y., Carrasco Kindb, M., & J. Brunner, R. (2017). Vizic: A Jupyter-based interactive visualization tool for astronomical catalogs. *Astronomy and Computing*, 20, 128–139. <https://doi.org/10.1016/j.ascom.2017.06.004>
- Wilkening, J. (2019). Towards Spatial Data Science: Bridging the Gap between GIS, Cartography and Data Science. *Abstracts of the ICA 2019*, 1, 1–2. <https://doi.org/10.5194/ica-abs-1-403-2019>
- Wu, Q. (2021). Leafmap: A Python package for interactive mapping and geospatial analysis with minimal coding in a Jupyter environment. *Journal of Open Source Software*, 6(63), 3414. <https://doi.org/10.21105/joss.03414>
- Yin, D., Liu, Y., Hu, H., Terstriep, J., Hong, X., Padmanabhan, A., & Wang, S. (2019). CyberGIS-Jupyter for reproducible and scalable geospatial analytics. *Concurrency and Computation: Practice and Experience*, 31(11), e5040. <https://doi.org/10.1002/cpe.5040>
- Zaharia, M., Chowdhury, M., Franklin, M. J., Shenker, S., & Stoica, I. (2010). Spark: Cluster computing with working sets. In *2nd USENIX Workshop on Hot Topics in Cloud Computing (HotCloud 10)*.