A Dynamic and Interactive Shiny App for Point Pattern Analysis

Specifically for Spatial Point Patterns Analysis and Network-Constrained Point Patterns Analysis

Jun Peng TEO
School of Computing and Information Systems
Singapore Management University
Singapore
junpeng.teo.2019@scis.smu.edu.sg

Yiling YU
School of Computing and Information Systems
Singapore Management University
Singapore
yiling.yu.2019@scis.smu.edu.sg

ABSTRACT

Point Pattern Analysis (PPA) is the study of the spatial arrangements of points in space. [1] A fundamental problem of PPA is to investigate whether the given spatial arrangement of points resembles random, regular, normal or clustered distribution. There are lots of methods for Point Pattern Analysis, which can be classified into two groups: density-based approach and distance-based approach. [2] Specifically, density-based measures investigate the variations of point densities across space while distance-based measures analyze the spatial distribution of points using distances between point pairs. [3] Researchers have been using these methods in various industries to generate insights, in the field such as epidemiology [4] and ecology [5].

There are many software and packages available online for researchers to do PPA, such as ArcGIS and Spatstat for R. However, not all researchers have adequate programing skills and knowledge to use the software and packages effectively. Without proper foundation, researchers will not have access to PPA methods, limiting the potential of research. Thus, we developed an R Shiny application, Spatial Pointers to assist researchers in conducting PPA. It integrates two of the most common analysis in PPA: Spatial Point Patterns Analysis and Network-Constrained Point Patterns Analysis. We designed Spatial Pointers in a way that is easy to use for people even do not have any programming knowledge. Just with a few clicks, users will be able to obtain high-quality interactive visualizations and interpretations for the two PPA analysis. In this paper, we will explain our approaches to PPA, our application design and illustrate the functions of Spatial Pointers using two use cases.

INTRODUCTION

Geospatial data is information that describes objects, events or other features with a location on or near the surface of the earth. [6] It usually integrates the geographic elements (the shape, size or location of the features) with the attribute information (the characteristics of the features). Countless data sources exist in the form of spatial data, ranging from census data, satellite image, weather data to social media data. According to the estimation by United Nations Initiative on Global Geospatial Information Management (UN-GGIM), 2.5 quintillion bytes of data is being generated every day, and a large portion of the data is location aware. [7] All these geospatial data can be analyzed to help professionals make strategic decisions in industries as diverse as banking, ecology, epidemiology, urban planning and supply chain management.

Point Pattern Analysis is one of the categories of such geospatial analysis, which is used to describe patterns of points over space (whether they have cluster, random or dispersed distribution), and making inference about the process that could have generated the observed pattern. [8]

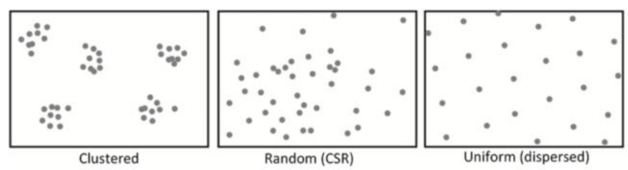

Figure 1: Three patterns of 256 points

To describe patterns of points over space, density-based method of PPA should be use. It concerns with the variations of point densities across the study area, which is a first-order property of the point pattern. To make inference about the process that could have generated the observed pattern, distance-based method of PPA should be use. It concerns with how the points are distributed relative to one another (attraction vs repulsion), which is a second-order property of the point pattern. [1] The density-based and distance-based methods are the fundamentals of PPA.

In the field of PPA, while some points of event/object/phenomena occur on a plane, some others occur on or alongside a network. We categorize the analysis of these two as Spatial Point Patterns Analysis (SPPA) and Network-Constrained Spatial Point Patterns Analysis (NetSPPA). A sample question under Spatial Point Patterns Analysis can be “Does the distribution of Airbnb listings are affected by location factors such as near to existing hotels, MRT services and tourist attractions?” A sample question under Network-Constrained Point Patterns Analysis can be “Do coffee outlets tend to stand side-by-side alongside streets in a downtown area?” Both Spatial Point Patterns Analysis and Network-Constrained Point Patterns Analysis are the focus of our application. For each Analysis, both first-order analysis and second order analysis will be performed by our application.

RELATED WORKS

Point Pattern Analysis has been conducted by researchers for many years. For example, Zhonghao Zhang, Rui Xiao, Ashton Shortridge and Jiaping Wu [9] conducted a case study on Human Settlements and Geographical Associations in Eastern Coastal China using PPA. Ripley's K function and Monte Carlo simulation of second-order analysis were used to investigate human settlement point patterns in the Wen-Tai region of eastern coastal China. In the study, all the PPA calculations and graphs were done by the Programita software. Programita software was developed by Dr. Thorsten Wiegand from Helmholtz Centre for Environmental Research – UFZ. [10] Though he is not a software engineer for commercial business, he recognizes the need for a tool like Programita for scientists to assist in research on PPA. Programita is a useful software allows scientists to perform second-order PPA analysis with Ripley's L-function and the O-ring statistic and also conduct Complete Spatial Randomness (CSR) tests on them. [11] However, since it was developed back in 2004, many functions and visualizations are considered as outdated nowadays. For example, though free, the software needs to be downloaded on a request basis, restricting its availability. Though interactive, the user interface design is not simple, intuitive and visual appealing and the users need to read a 166-page of user manual to fully utilize the software, limiting user experience. What's more, though it allows users to input data files, the files can only be in the format of *.dat and *.asc, which are outdated and not used many people in modern days.

In more recent years, there are also other resources online guiding scientists to perform PPA. For example, the book of Geographic Data Science with Python provides researchers with sample codes to conduct first-order analysis with Kernel Density Estimation (KDE), and second-order analysis with Ripley's G and F functions. [12] A limitation of such tutorial is that researchers need to have background in programming to understand and perform the codes. This requires users to devote lots of time in learning and thus limiting the efficiency of the tutorial.

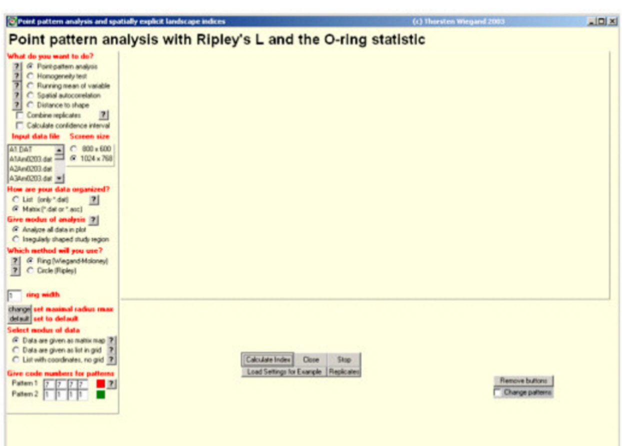

Figure 2: User Interface of Programita Software

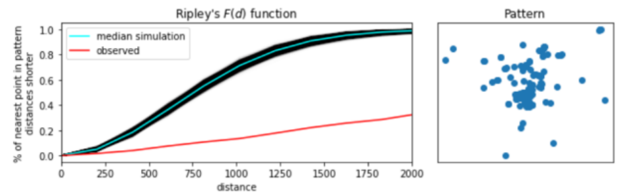

Figure 3: Sample Ripley's F function output using Python

All these limitations of currently available online free PPA resources bring us to develop our own application, Spatial Pointers.

MOTIVATION AND THE GOAL OF THE APPLICATION

Because of the common lack of effective and easy to use application on Point Pattern Analysis, we developed Spatial Pointers to assist researchers who do not have a programming background to explore in the field of PPA.

It aims to assist users, not limiting to any industry, to conduct either Spatial Point Patterns Analysis or Network-Constrained Point Patterns Analysis, without the need to learn programming languages.

Specifically, it attempts to achieve the following objectives:

1. To visualize the density of point patterns cartographically on internet-based maps such as OpenStreetMap
2. To create map visualisations that supports both macro and micro views
3. To conduct statistical testing that helps uncovers correlation patterns
4. To provide a user-friendly interface for users to use the application with ease

ANALYSIS APPROACHES BEHIND THE APP

Spatial Pointers provides users with both Spatial Point Patterns Analysis and Network-Constrained Point Patterns Analysis. For each of them, both first-order analysis and second order analysis will be performed.

1 Install R Packages

Since Spatial Pointers was developed using R, a free software environment for statistical computing and graphics [13], before diving into our analysis, we need to install the required R packages into the environment.

The first package to install is Shiny. We chose Shiny to build Spatial Pointers because it is a free, open source, extensible web applications framework for R that makes it easy to build interactive web interfaces. [14]

Other key R packages used in the PPA analysis include but is not limited to: sf for managing vector-based geospatial data, spatstat for performing first- and second-order PPA analysis, tidyverse for data wrangling and visualisation as well as tmap for plotting cartographic quality static or interactive point patterns maps. Figure 4 provides a detailed explanation for all the packages we used in our application.

Package	*Description*
shiny	Web Application Framework for R
tidyverse	To work with CSV files
tools	To ensure R packages work properly
shinythemes	To work with themes for shiny application
shinyjs	To perform JavaScript operations in R
tmap	To plot cartographic quality point pattern maps
sp	For working with spatial data
sf	To help manage vector-based geospatial data
rgdal	To help ensure correct CRS for objects using spTransform()
spNetwork	Provides functions to perform NetSPPA
spatstat	Provides functions to perform SPPA
raster	To convert spatstat output into raster format
maptools	Provides tools for manipulating geographic data

Figure 4: Description of R packages used in Spatial Pointers

2 Spatial Point Patterns Analysis

2.1 Data Wrangling/Preparation

Every Geospatial Analysis starts with managing input data. To conduct Spatial Point Patterns Analysis, the input datasets should include the event/object data points and the spatial boundary map of the geographical area that these event/object occur. The input data can be in the format of aspatial, such as comma separated values (CSV) with geo-coordinates of the object/event, or in the format of geospatial, such as Shapefile, KML or GeoJSON. These data will be imported by the functions such as read_csv() from tidyverse package or st_read() from sf package. Then, for imported CSV files, st_as_sf() from sf package is used to convert them to simple feature (sf) objects. After that, st_transform() and st_set_crs() from sf package will be used to ensure the sf objects are projected in same projection system and have proper Coordinate Reference System (CRS) information.

Because spatstat package requires the analytical data in point pattern dataset (ppp) object form, we will need to transform our data from sf objects to ppp objects. This will be done by three steps: converting sf objects to Spatial class, then to generic SpatialPoints or SpatialPolygons objects, lastly into ppp objects. Note that the point pattern may contain duplicated points, rjitter() function from spatstat package will be used to handle them.

When analyzing spatial point patterns, it is a good practice to confine the analysis with a geographical area like Singapore boundary. In spatstat, an object called owin is specially designed to represent this polygonal region. Thus, the imported spatial boundary map will be used to create such owin object which will be combined with the point event objects. By reaching this step, we will be prepared to conduct first-order and second-order point patterns analysis. Most of the work in this data wrangling section are done on the server side of the application and therefore hidden from the user.

2.2 First-order Spatial Point Patterns Analysis: Kernel Density Estimation (KDE)

To investigate the variations of point densities across space, density-based measures are being applied. Density-based measures can be divided into two categories, global density and local density. Global density is a simple and straightforward measure to see how many points on average per unit area across the whole study area. Local density, however, shows varying point densities at different locations in the study area. [3]

Kernel Density Estimation (KDE) is one of the mostly used density-based measures to estimate local density. It creates a grid which each cell is assigned the density value of the kernel window centered on that cell. The density value is estimated by counting the number of object/events in that kernel window. Examples of popular kernel functions are gaussian, uniform and quartic. Most kernel functions assign heavier weights to nearby points than distant points, producing a smoother density map.

Besides kernel functions, which specify the shape of points distribution in each kernel, the kernel bandwidth is another parameter of KDE, which controls the size of the kernel at each point. There are lots of choices of bandwidth, however, there is no simple recipe for choosing. A small value of bandwidth may result in the estimated density layer too jaggy while a large value of bandwidth may result in the estimated density layer too general and ignoring local features. Background information on the object/event that form the pattern should be considered in making choice of bandwidth.

KDE with choices of kernel functions and bandwidths can be performed by our application on SPPA.

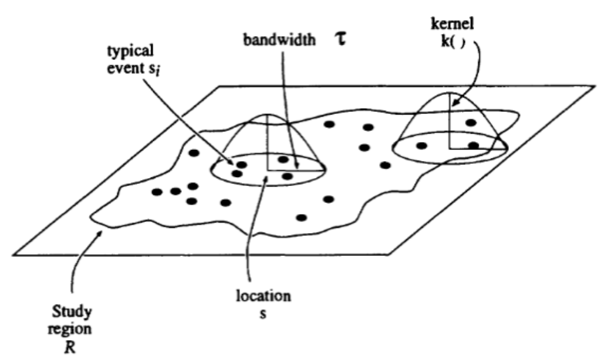

Figure 5: Kernel Estimation of a point pattern [15]

2.3 Second-order Spatial Point Patterns Analysis

To investigate the degree of spatial distribution of points or the interactions between points, distance-based measures are being applied. To describe the degree of spatial distribution, distance functions G-function and F-function will be performed by our application. To explore the interactions between multi-type points, Cross L-function will be performed by our application.

For each function, a Monte Carlo simulation test of Complete Spatial Randomness (CSR) will be performed. The null hypothesis is that the points are distributed randomly and there is no interaction between points. The alternative hypothesis is that the points are not distributed randomly which suggest they may be clustering or dispersion involved.

2.3.1 G-Function

The G-function calculates the cumulative frequency distribution of the nearest neighbor distance of a point pattern. It is written as:

$$G(d) = \frac{sum(D_{ij} < d)}{n}$$

Figure 6: G Function

where sum(Dij < d) stands for the number of point pairs i and j with a distance smaller than d, and n represents the total number of points. [3]

A Monte Carlo simulation test of CSR will be performed on G-function by our application. We need to compare the observed G to the theoretical G to see if the observed pattern is a likely realization of the hypothesized process.

The resulted pointwise envelopes in the graph specify the critical points for a Monte Carlo test. [16] If the observed G is inside the envelope, it means the null hypothesis of CSR cannot be rejected and we conclude the points resemble random distribution. If the observed G is above the envelope, we can reject null hypothesis and conclude the points resemble clustered distribution. If the observed G is below the enveloped, we can reject null hypothesis and conclude the points resemble dispersed distribution.

2.3.2 F-Function

The F-function first generates a few random points (denoted as P) in the study area, and then it determines the minimum distance from each random point in P to any original points (denoted as O) in the study area. The F function is written as:

$$F(d) = \frac{sum[d_{min}(p_i, s) < d]}{n}$$

Figure 7: F Function

where F(d) indicates the value of the F function at distance d, and sum[dmin(pi,s) < d] is the number of points in P with a minimum distance to any point in O smaller than d. [3]

The result interpretation of Monte Carlo simulation test of CSR for F-function is opposite of that for G-function. If the observed F is above the envelope, we can reject null hypothesis and conclude the points resemble dispersed distribution. If the observed F is below the enveloped, we can reject null hypothesis and conclude the points resemble clustered distribution.

2.3.3 Multi-type Point Patterns Analysis: Cross L-Function

To measure the interactions between multi-type points, Cross L function will be performed by our application. Cross L-function is a linearization of the Cross K-function which makes it easier to compare the observed and theoretical values.

Simply put, Cross K-function and Cross L-function measure the number of type A points up to a given distance from a type B point.

The result interpretation of Monte Carlo simulation test of CSR for Cross K-function and Cross L-function is still similar to G and F functions. If the observed L is inside the envelope, it means the null hypothesis of CSR cannot be rejected and we conclude the two types of points resemble random distribution and are independent of each other. If the observed L is above the envelope, we can reject null hypothesis and conclude two types of points resemble attraction patterns, suggesting clustering. If the observed L is below the enveloped, we can reject null hypothesis and conclude the two types of points resemble repulsion patterns, suggesting dispersion.

3 Network-Constrained Point Patterns Analysis

3.1 Data Wrangling/Preparation

Same as SPPA, NetSPPA data must be imported first.

For spatial data in shapefile format, readOGR() of rgdal package is used for importing and spTransform() of rgdal package is used to ensure accurate CRS.

For aspatial data in CSV format, read_csv() from tidyverse package is used for importing and st_as_sf() is used to ensure accurate CRS. After that, as() is used to coerce the imported aspatial object into spatial object, as spatial objects are required to run nkde() function of spNetwork.

3.2 First-order Network-Constrained Point Patterns Analysis: Network-Constrained Kernel Density estimation (NetKDE)

Network-Constrained Kernel Density estimation (NetKDE) is similar to KDE but used to calculate density of events occurring along the edges of a network.

NetKDE of variables with choices of kernel functions and method can be performed by our application on SPPA.

3.3 Second-order Network-Constrained Point Patterns Analysis

For Second-order Network-Constrained Point Patterns Analysis, Monte Carlo simulation test of CSR will be performed on K and Cross-K functions.

3.3.1 K-Function

Simply put, Ripley's K-function measures the number of point events found up to a given distance of any particular point event.

The result interpretation of Monte Carlo simulation test of CSR for K-function is the same as that for G-function.

3.3.2 Multi-type Point Patterns Analysis: Cross K-Function

See 2.3.3

THE APPLICATION AND DESIGN FRAMEWORK

This section presents a detailed description of the design principles followed, and application features built in our application.

At the topmost of the application is the Navigation Tab Bar, which shows the different functionalities our application can provide for users, through four selectable tabs.

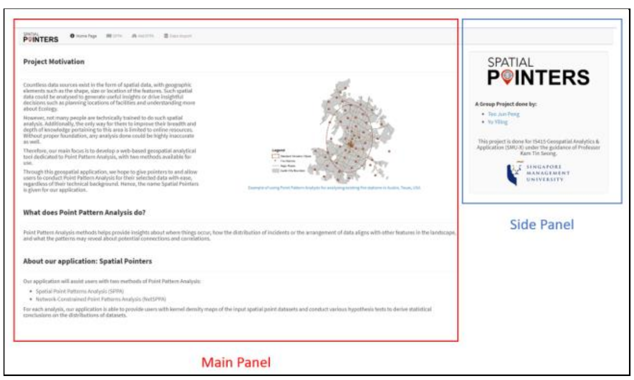

Figure 8: Standardized Layout for Application

Our design for our application closely follows a standardized layout for the various tabs: a main visualization panel on the left and a side panel for selection of inputs on the right. With this layout, it allows users to get a quick overview of the chosen tab at one glance (Fig. 8).

The Home Page tab will be the first tab that you can access in our application. From this page, users can have a quick overview of the motivation behind our application, what our application is all about and also gain a quick understanding of what Point Pattern Analysis can do for them.

The SPPA and NetSPPA tabs are the second and third tabs that you can access, with both tabs following the standardized layout mentioned previously. In the main panel, users can choose between the various sub-tabs: e.g. 1) Kernel Density Estimation, 2) K-Function. The map visualization tile in the main panel also creates an interactive experience for users with its Zoom Control and Base Map Control features. Users can zoom further in or out on the map and control the base map of the visualization to suit their visual interest. For the side panel located on the right, several input options are given for users to customize their visualization. In the case for Kernel Density Estimation, some examples of input selections include but is not limited to: 1) Variable for computing KDE, 2) Kernel method for computing KDE etc. For Statistical Functions, some input selections include but is not limited to: 1) Variable for computing G/F/K-Function, 2) An input box for inputting the Number of Monte Carlo simulations to be run etc.

The last accessible tab is the Data Import tab, where users can upload their own data in shapefile format. At least four types of files (.shp, .shx, .dbf, .prj) are mandated to be uploaded in order to ensure the data uploaded has the essential information it needs. Through this data import function, we hope to allow users to carry out PPA in our application using their own set of data.

The combination of the respective tabs creates a full-on experience for users and the opportunity for them to test out various input selections to suit their own use cases.

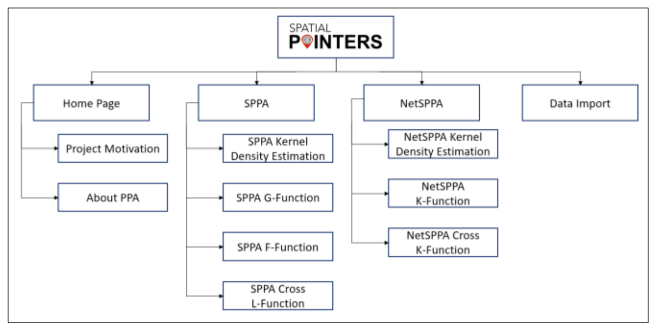

Figure 9: Structure of Spatial Pointers

DEMONSTRATION

1 Spatial Point Patterns Analysis

To illustrate the functions for SPPA, we would like to provide a user case for our application. The question to investigate is whether McDonald's outlets in Singapore are distributed randomly and if not, what are the factors that affect the outlets' locations. We imported data sets of McDonald's outlets and other 4 location factors in Singapore: KFC outlets, Gyms, MRT stations and Community Centers.

The result of Monte Carlo simulation test of CSR on G-function implied the McDonald's outlets resemble randomness from 0-0.4km and clustering pattern after 0.4km. The result of Monte Carlo simulation test of CSR on F-function implied the McDonald's outlets resemble randomness for all distances.

The results of Monte Carlo simulation test of CSR on Cross-L function suggest McDonald's outlets are not spatially independent with all 4 location factors from 0-4km and resemble attraction patterns. These results give insights to researchers how McDonald's strategically locates its outlets as the 4 locations factors may bring it customers and profits.

2 Network-Constrained Point Patterns Analysis

To illustrate the functions for NetSPPA, we inputted data sets of various point events (e.g. Childcare Centres) in Punggol, Singapore and several chosen secondary factors in the same study area. Through this, we will investigate whether the point events (e.g. Childcare Centres) in Punggol, Singapore is distributed randomly and if not, what are the secondary factors (e.g. Bus Stops) that affect their locations.

The result of Monte Carlo simulation test of CSR on K-function implied that Childcare Centres in Punggol, Singapore resemble regular/dispersion pattern from 0-400m and randomness after 400m.

The result of Monte Carlo simulation test of CSR on Cross-K function generally suggests Childcare Centres in Punggol, Singapore are spatially independent of all 3 location factors from 0-1000m. These results give insights to city planners on how Childcare Centres are located in Punggol, Singapore and how the locations can be further improved.

DISCUSSION

We developed Spatial Pointers, aiming to facilitate PPA for users not limiting to any industry, and assist them in conducting either Spatial Point Patterns Analysis or Network-Constrained Point Patterns Analysis, without the need to learn programming languages. By developing the application using R Shiny package, we were able to produce a quality web-based application that is accessible to everyone, with consideration given to the time frame of the project.

The user-friendly interface packed with substantial functionalities allows users to perform SPPA and NetSPPA without much issues. Since the application is intuitive and offers only the simplest explanations, users can interpret the outputs of various functions without having to research further, avoiding a steep learning curve. Having said that, users are recommended to read more about the explanations or the various input options in their free time, so as to further their knowledge in PPA.

We have also conducted App Users Testing with our friends informally, to gather feedback about our application. All of them agreed that the application user interface is clean and intuitive, and they were able to try out the various functions with ease. Moreover, they thought the explanations of the outputs are short and sweet and easy to understand. We are glad our users like the simplicity of our application.

FUTURE WORK

Firstly, a more generic data import function is coveted. Currently, our application only accepts data uploaded in shapefile format, which could be limiting for certain use cases. Hence, support for other types of data inputs can be added in future given the flexibility of R to accept many types of data. Automated data cleaning and quality control steps could also be implemented to complement the data import function, meaning that users would not need to preprocess the data before uploading, though certain guidelines would still be present to ensure data is imported correctly.

Secondly, additional PPA methods can be added to further expand the capability of our application in future. For first-order density-based measures, for example, we can add Quadrat Density Analysis. For second-order distance-based measures, for example, we can add Nearest-Neighbor Analysis.

Lastly, our application could be integrated with a real-time database so as to provide a more substantial demonstration for first time users of PPA. The database could contain various kinds of use cases, so as to help users understand the several types of use cases PPA can do for them.

CONCLUSION

In this paper we set out to present the thought process behind our web-based application Spatial Pointers and its development journey. Because of the lack of an effective and easy to use application on PPA, we developed Spatial Pointers to assist researchers who do not have programming backgrounds to explore in the field of PPA.

We designed the application interface in a way that is simple and intuitive to make it user-friendly. We also offer users the simplest explanations possible to make map/graph results easy to interpret and understand. The application allows users to perform SPPA and NetSPPA without spending time on researching or coding, serving as an effective exploratory tool for all users.

Accessibility wise, Spatial Pointers is an open-source web-based tool that is implemented using R and Shiny. This means that the application is freely available and all potential users from across the globe are welcome to use our application online.

To sum it all up, Spatial Pointers was created to address the lack of existing effective PPA oriented applications. We hope our application can be helpful for researchers to do PPA analysis.

REFERENCES

[1] Okabe, Atsuyuki, Barry Boots, and Kokichi Sugihara. 1992. *Spatial Tessellations: Concepts and Applications of Voronoi Diagrams.* DOI: https://academic.microsoft.com/paper/2092924973

[2] Manuel Gimond. 2021. *Intro to GIS and Spatial Analysis.* DOI: https://mgimond.github.io/Spatial/index.html

[3] Yuan, Y., Qiang, Y., Bin Asad, K., and Chow, T. E. 2020. Point Pattern Analysis. *The Geographic Information Science & Technology Body of Knowledge* (1st Quarter 2020 Edition), John P. Wilson (ed.). DOI: 10.22224/gistbok/2020.1.13.

[4] Gatrell, A. C., Bailey, T. C., Diggle, P. J., & Rowlingson, B. S. 1996. Spatial Point Pattern Analysis and Its Application in Geographical Epidemiology. Transactions of the Institute of British Geographers, 21(1), 256–274. DOI: https://doi.org/10.2307/622936

[5] Thorsten Wiegand and Kirk A. Moloney. 2014. Handbook of Spatial Point-Pattern Analysis in Ecology. DOI:https://books.google.com.sg/books?id=jwfLBQAAQBAJ&lpg=PP1&dq=point%20pattern%20analysis&lr&pg=PR4#v=onepage&q=point%20pattern%20analysis&f=false

[6] IBM. What is geospatial data?. DOI:https://www.ibm.com/topics/geospatial-data

[7] Jae-Gil Lee and Minseo Kang. 2015. Geospatial Big Data: Challenges and Opportunities, *Big Data Research, Volume 2, Issue 2, 2015, Pages 74-81, ISSN 2214-5796,* DOI: https://doi.org/10.1016/j.bdr.2015.01.003.

[8] Edzer Pebesma and Roger Bivand. 2021. Spatial Data Science with applications in R. DOI: https://keen-swartz-3146c4.netlify.app/

[9] Zhang, Zhonghao, Rui Xiao, Ashton Shortridge, and Jiaping Wu. 2014. Spatial Point Pattern Analysis of Human Settlements and Geographical Associations in Eastern Coastal China — A Case Study. *International Journal of Environmental Research and Public Health 11, no. 3: 2818-2833.* DOI: https://doi.org/10.3390/ijerph110302818.

[10] Programita. 2021. https://programita.org/

[11] Thorsten Wiegand. 2004. Introduction to Point Pattern Analysis with Ripley's L and the O-ring statistic using the Programita software. *A user manual with an collection of examples for point pattern analysis using the Programita software.* DOI:*http://ecovirtual.ib.usp.br/lib/exe/fetch.php?media=ecovirt:roteiro:manual programita2004b.pdf*

[12] Sergio J. Rey, Dani Arribas-Bel, and Levi J. Wolf. 2020. Point Pattern Analysis. DOI:https://geographicdata.science/book/notebooks/08_point_pattern_analysis.html

[13] R. 2021. The R Project for Statistical Computing. DOI: https://www.r-project.org/

[14] RStudio. 2020. Shiny from RStudio. DOI: https://shiny.rstudio.com/

[15] Anthony C. Gatrell, Trevor C. Bailey, Peter J. Diggle and Barry S. Rowlingson. 1996. Spatial Point Pattern Analysis and Its Application in Geographical Epidemiology. *Transactions of the Institute of British Geographers, 21(1), 256–274.* DOI: https://doi.org/10.2307/622936

[16] Rdocumentation. Spatstat (version 1.64-1). *envelope: Simulation Envelopes of Summary Function.* DOI:https://www.rdocumentation.org/packages/spatstat/versions/1.64-1/topics/envelope