AN EMBEDDED SPATIAL STATISTICS TOOLBOX IN OPEN SOURCE GIS SOFTWARE (uDig)

A THESIS SUBMITTED TO
THE GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCES
OF
MIDDLE EAST TECHNICAL UNIVERSITY

BY

MAHMUT ÇAVUR

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR
THE DEGREE OF DOCTOR OF PHILOSOPHY
IN
GEODETIC AND GEOGRAPHIC INFORMATION TECHNOLOGIES

FEBRUARY 2016

Approval of the thesis:

AN EMBEDDED SPATIAL STATISTICS TOOLBOX IN OPEN SOURCE GIS SOFTWARE (uDig)

submitted by **MAHMUT ÇAVUR** in partial fulfillment of the requirements for the degree of **Doctor of Philosophy in Geodetic and Geographic Information Technologies Department, Middle East Technical University** by,

Prof. Dr. Gülbin Dural Ünver ______________________
Dean, Graduate School of **Natural and Applied Sciences**

Assoc. Prof. Dr. Uğur Murat Leloğlu ______________________
Head of Department, **Geodetic and Geographic Inf. Tech.**

Prof. Dr. H. Şebnem Duzgun ______________________
Supervisor, **Mining Engineering Dept., METU**

Examining Committee Members:

Assoc. Prof. Dr. Uğur Murat Leloğlu ______________________
Geodetic and Geographic Inf. Tech. Dept, METU

Prof. Dr. H. Şebnem Düzgün ______________________
Mining Engineering Dept., METU

Assist. Prof. Dr. Serkan Kemeç ______________________
City and Regional Planning Dept., Yuzuncu Yil Unv.

Prof. Dr. Ahmet Coşar ______________________
Computer Engineering Dept., METU

Assist. Prof. Dr. Gülcan Sarp ______________________
Geography Dept., Suleyman Demirel Unv.

Date: 19.02.2016

I hereby declare that all information in this document has been obtained and presented in accordance with academic rules and ethical conduct. I also declare that, as required by these rules and conduct, I have fully cited and referenced all material and results that are not original to this work.

Name, Last name: Mahmut ÇAVUR

Signature

ABSTRACT

AN EMBEDDED SPATIAL STATISTICS TOOLBOX (R TECHNIQUES) IN OPEN SOURCE GIS SOFTWARE (uDig)

Çavur, Mahmut

Ph.D., Department of Geodetic and Geographic Information Technologies,

Supervisor: Prof. Dr. H.Şebnem Düzgün

February 2016, 236 pages

It is widely considered that geographic information systems (GIS) should include more spatial data analysis (SDA) techniques. The issues of which techniques should be included and how statistical analysis can be integrated with GIS are still widely debated. However, the typical software does not include all geospatial techniques. In this respect, this thesis focuses on the means to develop a framework which implements R spatial statistical techniques in the uDig GIS so that GIS and spatial statistical analysis can mutually benefit from such an integration. In this respect, this thesis introduces a framework that implements R spatial statistical techniques in the uDig GIS -. For this purpose, a simple interface is designed between two open-source software applications, uDig and the R statistical software package. The platform used here is Windows since the uDig and R compiles and installs smoothly, and they run in Linux and Mac operating system as well. The R techniques, tight integration strategy and RCaller are redesigned with respect to methodology requirements. In conclusion, the integration is successfully implemented and tested by users and developers. There are 14 spatial statistical and four non-geospatial techniques that have been integrated into uDig GIS softwareDevelopers can use proposed the framework to embed any required geospatial technique into uDig easily.

Keywords: A Framework to Embed Geospatial Techniques into GIS; Integration of Geospatial Techniques into GIS; A Tightly Coupled GIS with Geospatial Techniques; R Geospatial Techniques into GIS

ÖZ

MEKANSAL VE MEKANSAL OLMAYAN ANALİZ TEKNİKLERİNİN AÇIK KAYNAK KODLU BİR CBS YAZILIMINA ENTEGRASYONU (uDig)

Çavur, Mahmut

Doktora, Jeodezi ve Coğrafi Bilgi Teknolojileri Bölümü

Tez Yöneticisi: Prof. Dr. H. Şebnem Düzgün

Şubat 2016, 236 Sayfa

Mekânsal ve istatistiksel analiz tekniklerinin Coğrafi Bilgi Sistemleri (CBS) yazılımlarında daha yaygın kullanılması son yılların ilgi çeken araştırma konularındandır. Fakat hangi tekniklerin, nasıl entegre edileceği konusu halen tartışılmaktadır. Mevcut CBS yazılımları bütün mekânsal ve istatistiksel teknikleri barındırmamaktadır. Bundan dolayı, bu tez çalışması R tekniklerinin uDig GIS yazılımına nasıl entegre edileceğini bir sistematik ve metot ile her iki yazılımın da birbirinden faydalanacağı şekilde sağlamaktadır. Bu amaçla, açık kaynak kodlu ve ücretsiz olan R ve uDig GIS arasında ortak ve basit bir arayüz tasarlanıp geliştirildi. R ve uDig'in sorunsuz çalıştığı Windows platformu, geliştirme ve test amaçlı kullanıldı. Ayrıca entegrasyon, Linux ve Mac işletim sistemlerinde de başarılı bir şekilde çalıştırıldı. Tight entegrasyon metodu ve R Caller derleyicisi iki yazılım arasındaki iletişimi kurmak amacıyla kullanıldı. Kullanılan R teknikleri, R Caller derleyicisi ve

tight entegrasyon yöntemi ihtiyaçlara göre yeniden tasarlandı ve kodlandı. 14 tane mekânsal ve 4 tane istatistiksel analiz yöntemi bu tez kapsamında uDig yazılımına entegre edilmesine rağmen; geliştiriciler için Histogram Grafiği tekniğinin nasıl entegre edildiği açıklandı. Bu R tekniği için kullanıcı dostu bir arayüz tasarlanmış, başarılı bir şekilde tez kapsamında belirtilen sistematik ve metodoloji sayesinde uygulanmıştır. Ayırca, Çekirdek Yoğunluğu Tahmin Modeli, K Fonksiyonu ve Xls içe aktarma metotları, entegrasyonun başarılı bir şekilde çalıştığını göstermek amacıyla uygulandı. Sonuç olarak, entegrasyon sistematiği ve metodolojisi uygulanarak; R mekânsal ve istatistiksel teknikleri başarılı bir şekilde uDig yazılımına entegre edilip, son kullanıcılar tarafından test edilmiştir. Geliştiriciler, tavsiye edilen sistematiği ve metodolojiyi uygulayarak ihtiyaç duyulan herhangi bir mekânsal ve istatistiksel R tekniğini uDig GIS yazılımına kolayca entegre edebileceklerdir.

Anahtar Kelimeler: Mekânsal Tekniklerin CBS Yazılımına Entegrasyonu Sistematiği, Mekânsal Tekniklerin CBS Yazılımına Entegrasyonu, Tight Entegrayon Metodu ile Entegre Edilmiş CBS ve Mekânsal Analiz Yazılımı, R Mekânsal Tekniklerinin CBS’de Kullanımı

To My Dear Wife and Lovely Son

ACKNOWLEDGMENTS

I would like to express my sincerest appreciation to my supervisor Prof. Dr. H. Şebnem Düzgün for her continuous guidance and patience throughout this study. Studying with her provides me to see many valuable things and perspectives throughout my university education. She redounds me not only technical value but also ethical value on my life. In brief, honesty that I learned from her is the best title and value in the rest of my life.

I would like to express my special thanks to my committee member Assoc. Prof. Dr. Uğur M. Leloğlu for his continuous support throughout this study. I feel very lucky to be acquainted with his wisdom. I would also like to thank my committee member Assist Prof. Dr. Serkan Kemeç for his valuable suggestions and support. The contributions of my committee members, Prof. Dr. Ahmet Coşar and Assist Prof. Dr. Gülcan Sarp are also gratefully acknowledged.

I am deeply thankful to my best friends Mahmut Camalan and Ebübekir Demir, who were beside me especially during the last steps of my thesis. I would also like to thank all my other friends and colleagues for their friendship and support, particularly Habip Demir, Erbil Postallı, Metehan Demir, Muzaffer A. Sezen, Mehmet Karaimam, Ekin Güngör, Muhammed Ersönmez, Mehmet Çelik, Halil Tutal, Bilge Çelik, İlay Çelik, Ahmet Süzen, A. Bera İçli, Zemzem Taşgüzen, and Damla Erdoğan.

Last but not least, I wish to express my heartfelt thanks to my sisters and brothers, my father-in-law, Erhan Erdoğan; mother-in-law, Nezahat Erdoğan; my father, Ahmet Çavur and mother, Zeliha Çavur for their incredible understanding, pure love, great patience and emotional support during my PhD study. Specifically, I am thankful to my wonderfully supportive wife for her endless love, motivation, understanding, encouragement and patience throughout my study.

TABLE OF CONTENTS

LIST OF TABLES

TABLES

LIST OF FIGURES

FIGURES

LIST OF ABBREVIATIONS

SDA	Spatial Data Analysis
GIS	Geographic Information System
FLOSS	Free/Libre and Open Source Software
QGIS	Quantum GIS
ASCIII	American Standard Code for Information Interchange
GUI	Graphical User Interfaces
UI	Uniform Intensity
KDE	Kernel Density Estimation
AKE	Adaptive Kernel Estimation
uDig	User Friendly Desktop Internet GIS
KST	Kolmogorov-Smirnov Test
FPM	Fit Poisson Model
UDE	Univariate Density Estimation
GWR	Geographical Weighted Regression
SR	Spatial Regression

CSR	Complete Spatial Randomness
ESRI	Environmental Systems Research Institute
SDI	Spatial Data Infrastructures
GRASS	Geographic Resources Analysis Support System
GPL	General Public Licence
GNU	Recursive Acronym for GNU is Not UNIX
CSISS	Center for Spatially Integrated Social Science
MCDA	Multi-Criteria Decision Analysis
GIScience	Graphical Information Science
OGC	Open Geospatial Consortium
FOSS	Free/Open Source Software
SDLC	Software Development Life Cycle
API	Application Programmers Interface
EDA	Exploratory Data Analysis
XML	Extendable Markup Language
RS	Remote Sensing
OS	Operating System

CHAPTER 1

INTRODUCTION

Geographic Information System (GIS) and Spatial Data Analysis (SDA) are two complementary software packages used to meet the requirements of the GIS and SDA community. Both communities require different features of these two packages for different purposes. In this chapter, first the motivation is explained in order to integrate these two packages. Then the aim of the study is clearly defined and explained. After that, the main contribution of this study to GIS and SDA community is explained. Finally, the organization of the thesis is explained in the last section of this chapter.

1.1.Motivation of the Thesis

There are increasing numbers of GIS software containing several SDA techniques in the market for the usage of GIS community. However, the number of GIS including SDA techniques and the number of SDA techniques integrated into GIS are not enough for users. Since more advanced tools for spatial analysis are preferred nowadays, it will be needed to have such integrations as sooner as possible (Kepoglu, 2011). This expectation increases the desire to develop new SDA techniques integrated into GIS software by applying the Free/Libre and Open Source Software (FLOSS) development methodology. In addition, the developers need a straightforward framework and methodology to develop a new integration for the users. Currently, there is no straightforward strategy for developers to integrate any required SDA technique into GIS software. It means that one should start to develop a new methodology before the application of a new technique into a GIS software. This prevents developers and researchers from making a new integration and taking initiatives. Because of those two main reasons, the GIS and SDA should be integrated with a framework that handles the integration in an easy, applicable and straightforwardway. SDA requires the components and power of GIS in visualizing, exploring, analyzing and estimating the behavior of the spatial processes, whereas GIS requires the capabilities of SDA to analyze the geospatial data. The success of any GIS system can be achieved by providing effective interaction between these components, enabling knowledge to be gained. Even if one component of GIS does not function completely or meet the requirements of the user, it will cause certain undesirable problems. Therefore, an inadequate component of GIS should be complemented in different ways by the developers. Most of the SDA software are not powerful to visualize the data and the result of analysis. The visualization of geospatial information is a key issue for effective decision-making (Kemec and Duzgun, 2006). In addition to visualization, SDA and GIS are very important for each other to complement their weak features (Goodchild, Haining & Wise, 1992). Contrary to the requirement of SDA for GIS, the GIS also requires several features of SDA like the analysis power. Analysis techniques in SDA increase the power of GIS in order to gain more information from the data (Palmer, Bailey & Gatrell, 1996).

The most of the tools which contain advanced SDA techniques require script development or code development. Therefore, it is really difficult for users and developers to develop code or scripts every time for every process. Kepoglu (2011) stated that GIS users become accustomed to use menu driven SDA and GIS software recently. These types of works provide a visual interactive interface and make it easier for the user, yet there is no straightforward strategy for the developers to integrate any required SDA technique easily. Besides, there are not enough GIS software including SDA techniques for users. As a result, these factors strengthen the motivation to take the initiative for such a study.

The currently available GIS software should be more flexible and interactive for the users to understand and use it easily. Moreover, the visualization of GIS software adds valuable features and manipulation flexibility for the users. Finally, and most importantly, GIS users require more advanced geospatial data analysis tools to analyze and understand complex spatial phenomena clearly. To understand and manipulate the result, the output view is critical and important for the users which needs more interaction. To achieve such an integrated and complementary solution for SDA and GIS, a framework should be provided to meet the requirements of the users (Cremers et al., 1995). GIS is actually the outcome of integrating various technologies to the specific needs of spatial data. In this respect, the development about this field will be continuous due to the absence of required technologies. Therefore, every effort about this fiels will be likely to provide new technologies related with GIS. Then, it will stimulate new analytic methods and hypotheses about GIS. Although, the development that started in the 1960s, it has not been completed yet so as to meet the requirement of GIS and SDA community.

The process of stimulus and convergence that began in the 1960s with GIS and SDA is far from complete, and the interaction between them is likely to remain interesting and productive for many years to come.

However, the currently available software, independent of type, strategy and methods, has an insufficient SDA toolbox to meet the requirements of users in regard to visualization, manipulation, flexibility and the number of SDA techniques. Furthermore, developers experience certain problems in integrating any SDA technique into GIS due to lack of a standard strategy. This situation presents several problems such as costly conversion, time consuming usage and indentification of user needs. The users try to utilize different type of software for different type of analyses. Moreover, developers really have difficulties in finding the right SDA techniques and GIS software and related development environments and programming language(s). Selecting those components and their integration still remains as an unsolved problem for the developers. Anselin (1992) claims that a better integration methodology would be beneficial for the entire field of GIS. As noted by the authors, one of the key functions of GIS is the analysis portion, which in turn encompasses the spatial statistical analysis. They correctly identify this function as vital for more complex and in-depth case studies in the future.

Therefore, to overcome these problems, there is a need for integrating spatial statistical toolboxes with GIS softwarein order to reach all users in cost effective way. In order to meet this requirement, a unique, easy-to-apply and sustainable framework and methodology for developers is essential in order to embed any spatial statistical toolbox into a GIS software.

1.2.Aim and Scope

The main objective of this thesis is to provide a framework for developers to integrate any required spatial statistical toolbox into a GIS software. The framework has five steps. Each step is clarified and explained with case studies. Two different types of stakeholders have been targeted: the first one is the users, and the second one is the developers. For the users, an easy-to-use, menu-driven, and user-friendly GIS software program is provided including several SDA techniques. In this way, the users can use the integration without the knowledge of codes with a simple and user-friendly graphical user interface (GUI). For the second group, the developers, a framework and a related methodology are provided in order to integrate spatial statistical and non-spatial techniques into the GIS software. The framework and the related methodology decrease the workload of developers to integrate any required technique into the GIS software without spending much effort.

The framework involves the integration of R (open-source tool library for spatial and non-spatial analyses) with uDig (User-Friendly Desktop Internet GIS) by using the RCaller bridge technique, which entails a basic interface between the two environments. Although the framework developed for the Windows operating systems, it also runs on Linux and Mac OS successfully. The proposed framework is implemented for 14 different SDA functions—Uniform Intensity, Kernel Density Estimation, Adaptive Density Estimation, G Function, L Function, F Function, K Function, Kolmogorov-Smirnov Test, Simulation Envelope of CSR, Fit Poisson Model, Kriging, Geographical Weighted Regression, Spatial Regression and Density Estimation—and four non-spatial functions—Histogram Graph, Column Graph, Pie Chart and Statistical Summary tools.

A case study is explained through its source codes so as to prove the application of each step of the framework. Then, three case studies are applied with sample data to indicate the success of the framework and the study.

In addition to integrating the spatial statistical toolbox into GIS, several tests are applied with different sample data sets and different tasks. For this purposes, the alpha and beta tests are applied. The alpha test is applied by the developers of the study and the beta tests are applied to the users. If the alpha test is successful, then the beta test is applied with different tasks and sample data. In addition to the tests, a survey is conducted on the users after carrying out the defined tasks to get feedback from them and to understand their satisfaction levels. Finally, the hypotheses which are defined and explained to test the success of the study are analyzed. The result of the analysis is reported in order to prove the success of the study and for further research in the future.

1.3.Main Contributions

This study provides several contributions towards the SDA and GIS community. The most significant originality of the study is behind the proposed framework and the related methodology. The application of this framework not only integrates a specific spatial statistical technique/software into a GIS software but is also applicable and suitable for any spatial statistical technique/software in any GIS software. In addition to the framework and the methodology, there are several other reasons why this study is unique and valuable.

The first one is that the integration is installed and run on all OS's (Windows, Linux and Mac). In this way, all the GIS users, whether they are accustomed to using Linux, Window or Mac, can set up this integrated software application and use it. Most of the SDA software types have certain limitations to be set up on different operating systems (OS's). This decreases the usage of SDA software and after a while it causes the SDA software to be outdated. The proposed framework and the methodology can be used for 32- and 64-bit OS's. The variations in the product increase the usability and dissemination of the study as well.

Secondly, the proposed solution runs independently with regards to GIS versions. There are many studies in the past which are outdated due to version dependency (e.g. Kepoglu 2011). Therefore, the developed methodology adopts all the GIS versions currently available.

Third, the integration of the Java language and the R Scripts is a new case for researches. There are some examples of SDA being integrated into GIS in the past. The sda4pp was integrated as an extension to the Quantum GIS (QGIS). The Phyton and C languages were used for this integration. Since there is no such integration in the literature, it will meet the requirements of specifically the developers and researchers who are accustomed to using these programming language(s).

Fourth, one of the unique features of the study is that it allows to redesign the currently available software for the suitability to accept Excel, Access and ASCII type of input. The input/output type of the data and the results are taken into account while developing the framework and methodology. Importing/exporting Excel, Access and American Standard Code for Information Interchange (ASCII) types of data into the GIS environment is required for many users. The shapefile format introduced with the ArcView GIS version 2 in the early 1990s was saving a geometric location and an attribute as a vector, but could not store topological information. However, recently it has become possible to read and write geographical datasets using the shapefile format with a wide variety of software. Shapefile is accepted as a standard for the GIS software throughout the world but some of the users have different type of data in hand. Designing GIS software with respect to user requirements increase the usage and dissemination of the developed software by the GIS community. Therefore,

Fifth, there is no ready-to-use RScript for any type of SDA techniques discussed above. Each R Package and its script is unique and examined and tested with respect to the requirements. For this purpose, the RScript of each SDA technique, which is used in this study, is first created and tested on the R environment and then is integrated into the uDig using the RCaller methodology. Therefore, the RScript developed in the content of this study is called the “Structured RScript”, as another contribution

Sixth contribution is that GIS and SDA are two main parts that should be integrated with a methodology but there should be a linkage between these two software systems to be created for a complete and trouble-free communication. For this purpose, a free, open-source and designable bridge technique is selected and applied successfully. Therefore, the selected bridge technique is also called as the “Structured Interpreter”. Seventh is that there are four types of coupling applied mostly to integrate different software (Karimi & Houston, 1996). However, the selected coupling strategy also is redesigned and restructured with respect to integration, software and users' requirements. Therefore, it is called as the “Structured Coupling Strategy” like the RScript and interpreter.

Last but not the least, the source code of the integration is open-source. The source code of open-source application should be distributed and executable binaries should comprise the software itself (Heron, Hanson & Ricketts, 2013). The source code is open-source and distributed through a web site (www.sda4udig.com) which is designed and created specifically for this study. In this way, it is expected that the distribution and dissemination process be more practical. The development is more sustainable and it is designed with respect to feedback coming from the users who download the integrated software. All these contributions accelerate the dissemination and sustainability of the system.

1.4.Organization of The Thesis

The thesis includes six chapters that cover the corresponding subjects in an organized manner. A brief description of each chapter is as follows:

- *“**Introduction**” is chapter one and defines the motivation, aim and scope. In addition, it explains the problem statement and main contribution of this study.*
- *“**Literature Review**” is chapter two and emphasizes the importance of the SDA and GIS integration. It explains the overview of GIS, SDA and the integration of SDA with GIS.*
- ***“Integration of SDA with GIS Software”** is chapter three and explains the Coupling Strategies and Overview of Existing GIS and SDA Software.*
- ***“The Proposed Framework”** is chapter four and this is the most technical chapter to define and explains the framework and the methodology.*

Finally, the methodology is applied with respect to "Software Development Life Cycle (SDLC)" engineering rule is explained at the end of the chapter.

- ***"Application of the Proposed Framework"*** *is chapter five and explains the implementation of framework and the usage of three implemented techniques with their results. In addition, the Proposed Methodology is explained with a case study for developers. Finally, the developer and users' test and the result is explained in this chapter.*
- ***"Conclusion and Further Recommendations"*** *summarizes this study with discussion, further research and conclusion.*

CHAPTER 2

LITERATURE REVIEW

The integration and development of Spatial Data Analysis (SDA) with GIS started just after the development of GIS began. Although there are various types of development strategies and requirements, both the strategies and requirements have evolved with respect to the requirements of users and developers in time. Also the strategies depend on whether the initiative part is a governmental organization or from the private sector. In fact, the demand for SDA technique increases the research about the integration of SDA and GIS software. For those reasons, the integration has been achieved in different ways such as developing scripts, extension, plug-in, package, toolbar, module, add-on and development libraries. SDA techniques and tools have spread independent of the developmental strategies, because each development strategy has advantages and disadvantages. Some of them have been continuously developed in an open-source environment and some of them as a black-box strategy. This affects the continuity, sustainability, usage and diffusion of the study as well. The development process mostly depends on the software adequacy and the initiative of the developers. Therefore, there are various research and development studies in the literature explained in this chapter in greater detail.

2.1. Development of GIS and SDA

GIS is a widely-used computerized system to take, store, manipulate, analyze, manage, and present all types of spatial, spatiotemporal and geographical data. The major processes of a GIS system can be defined as data acquisition/input, management, manipulation, analysis and display (Figure 1).

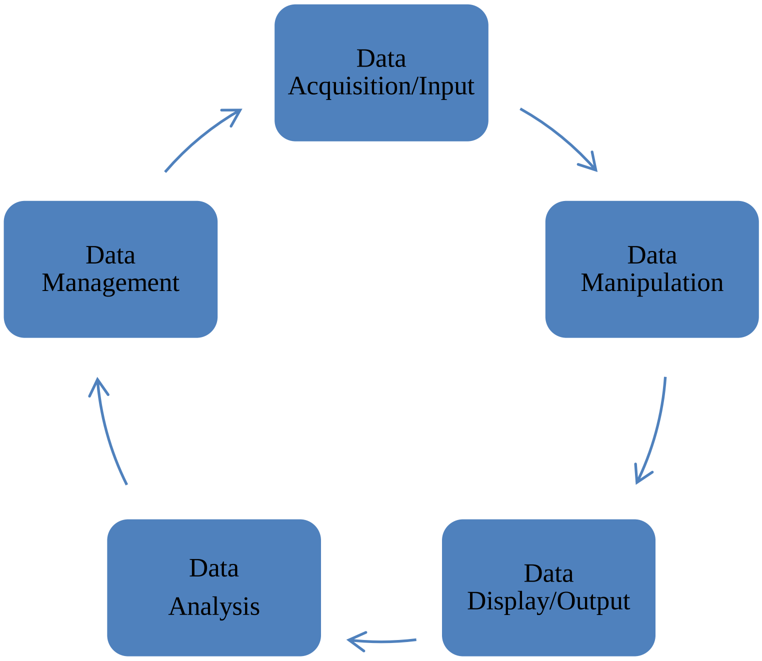


Figure 1. Components of GIS (adapted from Murray, 2009)

Before explaining the SDA techniques, the GIS components should be clarified to understand each component and their functions in order to define appropiate GIS and SDA integration.

Data acquisition/input makes geographic information readable and editable in a digital environment for further analysis. There are various types of data used by the GIS community. The shapefile format is one of the standards for GIS as an input. However, today the users who do not volunteer to convert data want to use different kinds of data formats, such as .mdb, .xls, .csv and ASCII file.

Data management provides efficient storage for and quick access to spatial information and its corresponding attributes. This task is achieved through several databases such as Oracle, MySQL, DB2 and PostGreSQL. Non-expert users can handle this task by using classical external or internal hard drives.

Display is the generation of map-based graphics from GIS data for further evaluation and inspection by the users. For a successful display, GIS added the wow factor to this method. The mapview of GIS should be as flexible as possible to provide any kind of manipulation for users.

Analysis consists of complicated, tedious subsets of actions, depending on the user. Basic analysis features of GIS are query (non-spatial and spatial), map algebra, buffering, and computational geometry functions like creating a voronoi diagram, point interpolation, histogram, and summary statistics. However, in recent years the users have required to use more advanced analysis techniques in order to increase the the interpretation of spatial processes for decision making and other related activities.

On the other hand, SDA is explained as mathematical techniques using attribute and spatial features of geographical and topological data to increase the understanding of them and make some predictions for the future or for unknown data. Goodchild (1987, cited in Goodchild, Haining & Wise, 1992) refers to SDA as a set of techniques devised to support the spatial perspective on data. Goodchild, Haining and Wise (1992) emphasize the distinction of this field that SDA techniques "are dependent on the locations of the objects or events being analyzed". Therefore, SDA techniques should require accessing to "both the locations and the attributes of objects". Bailey and Gatrell (1995) explain that the objective of SDA is to increase the basic understanding of the spatial process, to assess the hypotheses and to predict unknown values.

Recent developments in GIS and SDA software have made it possible to integrate spatial techniques to obtain a wide range of GIS operations. The need for advanced GIS and SDA techniques provide an opportunity for the evolution of both GIS and SDA tools and techniques. Therefore, the traditional SDA techniques have been insufficient to address the challenges faced in GIS and SDA software. SDA requires the components of GIS (Figure 1), which are input, display, analysis, manipulation and output, in order to overcome the challenges caused by the deficiencies of its tools. On the other hand, SDA has various types of SDA techniques for different types of needs. These requirements cannot be completely met by currently available GIS tools all the time. Therefore, both GIS and SDA complete the deficiencies of each other in different ways. Before talking about the requirements of SDA software, the definition of SDA should also be explained clearly.

SDA manages the circumstance where observational information is accessible on a few methodologies working in space and techniques that are tried to portray or clarify the conduct of this procedure and its probable relationship to other spatial phenomena. The main reasons behind the analysis are,

- Enhancing understanding of the spatial processes
- Explaining the spatial processes through various hypotheses
- Predicting values in areas where there aren't observations

(Bailey & Gatrell 1995).

SDA is an element of spatial analysis, consisting of a set of techniques and rules whose results are dependent on the locations of data being analyzed. Therefore, a successive SDA should access both the locations and attributes of data. It should contain simple statistical techniques and more advance spatial data analysis techniques. Some of them can be achieved by SDA techniques while some are achieved by GIS components. Goodchild (1992) defines SDA as the collection of methods created to provide a spatial background for the data. SDA represents the techniques and models where spatial referencing is used explicitly in different data cases. SDA is supposed to make assumptions on data and in this manner it should explain spatial relationships or interactions among cases. Furthermore, although it started to be regarded as a separate concept, SDA has a significant place in today's GIS applications. SDA will be developed further with a close examination of SDA and GIS applications (Goodchild, Haining & Wise, 1992). SDA originated separately from GIS, yet the former has been converted into the latter over the past 40 years. As science moves into a new era of technology-based collaboration and cyber infrastructure, it is clear that GIS and SDA should be complementary for each other. In analogy, their relationship could be resembled to the one between statistical packages and statistics, or word processors and writing. Clearly, these two perspectives can stimulate each other's process. In other words, GIS satisfies the demands of SDA, and SDA contributes to the focus which GIS brings for issues of representation and ontology. Moreover, GIS can become a medium of communication with SDA, where the latter is one of several ways for enhancing the message.

Typically, map definition and manipulation process are defined as spatial analysis by private sector (Anselin, 1992). Fischer and Nijkamp (1992) point out the importance of spatial modeling and statistical analysis in the review. In addition to GIS functions, the analysis function is also divided into four components (Anselin & Getis, 1992). These include selection (sampling of data), manipulation (partitioning, aggregation, overlay, buffering, and interpolation), exploration and confirmation (Anselin, 1992). It is clear that the selection and manipulation of data has been achieved by the currently available GIS software but the exploration and confirmation is the missing part of GIS and recently it has been tried to be solved in various ways. In Anselin, Dodson and Hudak (1993), it is claimed that the selection and manipulation functions are among the features of GIS but exploration and confirmation are defined as data analysis module. Therefore, the linkage between GIS and SDA is taken in much of the literature, which is often approached in terms of a linkage between two different software systems. To do that, numerous research studies have proposed various methodologies to integrate SDA and GIS features, such as a classification into close or loose coupling (Goodchild, Haining & Wise, 1992), or into encompassing and modular (Anselin and Getis, 1992). This integration has been for the advantage of both GIS and SDA over the past decades in order to increase the productivity of them (Fotheringham & Rogerson, 1994; Goodchild, 1988; Goodchild, Haining & Wise, 1992). Since there are two different sides to be integrated, GIS is used as the key point for SDA, making it more accessible for GIS and spatial data users, and hopefully it is more widely used to make effective decisions and to support scientific research (Goodchild & Longley, 1999).

Developing and applying manipulation, analysis and models to understand the real world is the main field of geographical data analysis. In this way, one can create new data and new information with the help of GIS and SDA tools. GIS software improves the processes by creating new information and inter-relation between data to increase better understanding of the real world. At this point, SDA completes the insufficient functionality of the GIS in order to increase the interpretation of spatial data. Therefore, SDA is a vital part of GIS. The reason of this symbiosis is that SDA creates and extracts new information about geospatial data through examination, evaluation, analysis and modelling of these data.

Spatial analysis, on the other hand, is a process of modeling, examining, and interpreting model for estimating and predicting, and for interpreting and understanding geospatial data (Encyclopedia, 2016). Spatial analysis in GIS contains attribute query, spatial query and new data generation (Bwozough, 1987 as cited in Raju, 2003). Raju (2003) also defines the scope of the spatial analysis as querrying spatial data, querrying attribute data and manipulation of data.

The different data types and related problems for SDA are summarized in Table 1.

Table 1. Types of Data and Problems in SDA

	Types of Data	**Example**	**Problems**
Point patterns analysis	Localized events	Trafic Accidents	Determination of patterns and aggregations
Surface analysis	Field data	Mine deposits	Interpolation and uncertainty
Areal analysis	Polygons	Census data	Regression and joint distributions

2.2. Overview of SDA and GIS Integration

The development of GIS and incorporating certain required simple SDA techniques into GIS started in the 1960s. In those years, the GIS community required spatial operations such as merging, cutting, overlay and intersections. These kinds of tools were adapted to GIS software as standard tools. After that, all the GIS software was developed, including with these tools and techniques, but the development of SDA progressed slower than that of GIS. Fotheringham and Rogerson (1993) also stated that SDA had a slower development than GIS techniques because of the priorities of requirements. The slower development and lack of integration caused SDA and GIS to spread slowly so far. By the early 1970s, researchers had realized the beneficial outcomes of geographic information using integrated software. The realization was followed by the ease in adding new geographic functions and capabilities with inconsiderable programming efforts (Goodchild & Haining, 2003). It is clear that this type of requirement emerged at the beginning of the 1970s just after the GIS was established. Then, the demand for SDA was specific with respect to the requirements of various disciplines. However, GIS vendors provided tools and techniques to the governmental organizations because of economic reasons (Fotheringham & Rogerson, 1994). The particular need in SDA software development began in the late 1980s (Haining, 1989). Since most scientists and users demanded such tools in the late 1980s, the research and the development for specialized software for SDA witnessed a rapid growth and was cited as a major impediment to the adoption and use of spatial statistics by geographic information systems researchers. Initially, researchers tried to understand the integration strategy, which spatial data analysis toolbox should be integrated and which coupling methodology is the most useful and reliable (loosely versus tightly coupled, embedded versus modular, etc.) (Anselin, Syabri & Kho, 2006). Anselin (1992), Goodchild, Haining and Wise (1992), Fotheringham and Rogerson (1993), and Anselin, Syabri and Kho (2006) focused on the necessity of computer software as well, in which SDA could be used in complete integration (Kepoglu, 2011). Though the integration started at the beginning of the 1990s, the integration was not adequate because different requirements were claimed from different disciplines, like biology, geography, statistics, remote sensing, computer sciences and mathematics. The Environmental Systems Research Institute (ESRI) is one of the most important commercial GIS software producers and started to develop SDA tools for different areas, yet there are not enough tools and techniques to meet the requirements within the products of the ESRI either. It is stated that ESRI ArcInfo licensed ArcGIS Desktop Version 10.0 has just 1.01% SDA tools (Kepoglu, 2011). After the acceptance of SDA with GIS tools, point in polygon, buffer, slop, intersection, summary statistics, etc. became the tools of GIS software because of their strong requirements for specific purposes. However, the currently available SDA techniques of GIS software are not adequate for the purposes of geospatial industry. Therefore, the potential of GIS and SDA can be achieved with a combination of them.

GIS and SDA need each other in terms of their compatible features. In addition to integrating some general SDA techniques into GIS software, there have been several specific efforts to integrate the specific SDA toolboxes into GIS especially in recent years. Since the 2000s, both the GIS community and spatial data users require a SDA integrating a GIS in order to use spatial data in an efficient and flexible way. These requirements emerged at the beginning of the 2000s since GIS was used by multidisciplinary community for various purposes (Hégron, Bocher & Petit, 2012). It means that the GIS and SDA community need a complementary solution containing the features of both SDA and GIS and more importantly, they need a straightforward framework or a methodology to implement any required SDA technique into a GIS software.

The need for SDA software development started in the late 1980s (Haining, 1989). Anselin (1992), Goodchild, Haining and Wise (1992), Fotheringham and Rogerson (1994), and Anselin, Syabri and Kho (2006) focused on the necessity of software in which SDA could be used in full integration with GIS. Although several methodologies for integrating SDA in GIS were started in the early 1980s, the first attempt was based on FORTRAN source code (Goodchild, Haining & Wise, 1992). Although statistical tools such as SAS and SPSS were developed in the early 1970s, the spatial modules did not satisfactorily comply with the GIS users. Although SAS and SPSS types of statistical tools were developed at the beginning of the 1970s, the spatial modules did not satisfactorily comply with the GIS users. Finally, it is emphasized that most of the techniques of SDA constituted complex and difficult functions, which required a different approach from the conventional approach of GIS (Goodchild, Haining & Wise, 1992). Recently, statisticians have dealt with the exploratory methods for geospatial data, including some of the characteristics such as trends, patterns, and spatial interactions. Therefore, one cannot explore the data merely with GIS and SDA tools. Trends, patterns, and spatial interaction can be understood via the visualization of GIS, while statistical summary and other spatial characteristics can be achieved by the SDA tools. Besides, Goodchild states that there are four areas where statistical developments might strengthen the present GIS practice: data rectification, data assessment, data sampling and initial data exploration (Goodchild, Haining & Wise, 1992). Therefore, the development of SDA tools will add more values and capabilities to GIS in the near future. To sum up, the development of SDA and GIS started so long time ago and will continue from now on with different methodologies and strategies.

Since the SDA and GIS benefit from each other and will complement their features in certain ways, their spread, usage and diffusion will support the development process in different strategies. GIS applications serve with their input, management, editing, and displaying features for SDA packages. Therefore, it is a vital component for the cases of utilizing GIS bundles as a vehicle for SDA (Haining, 1989). The usage and the development of GIS, which will be a common technique in handling spatial data, should resolve any case beyond data management further to data analysis (Goodchild, Haining & Wise, 1992).

Not only governmental organizations, but also private sector emphasizes the importance and necessity of the SDA and GIS integration. Longley (2001) approaches the significance of this integration from an economic aspect and focuses on that the economies of scale that underlie the viability of business GIS force an alternate critical limitation: the need to address the prerequisites of numerous applications all the while, and to give the best consideration regarding the biggest fragments of the business sector. SDA may be the most complicated and challenging of GIS applications; however, it is in no way, shape or form the most noteworthy one in commercial terms (Longley, 2001; Goodchild & Haining, 2003).

Although the requirements all imply the significance of SDA and GIS integration, there is little effort to come up with such integrated tools and techniques. As there are fewer discussions on the subject of the integration of SDA and GIS, only a small progress has been made until now. Resource management, infrastructure, facilities management, and land information have been dominating any discussion and the concept about the integration of SDA and GIS for years. Despite its promises, SDA is still a relatively ambiguous field, as it requires expertise in both spatial phenomena and statistics, as well as available recources (Goodchild, Haining & Wise, 1992). Therefore, the usage, spread and diffusion of the SDA and GIS integration also depends on available resources and the efforts of researchers.

The integration of SDA on GIS can be achieved by experts and software engineers, and also by the researchers. There is an inclination towards courses on the basics of GIS and SDA in many universities (Heywood, 1990; Goodchild, Haining & Wise, 1992). These types of academic organizations, i.e. conferences, lectures, meetings, seminars, and academic studies, will help develop SDA methods in GIS and will increase the desire for an integrated software. In addition, as it was discussed in more detail before, an open-source strategy will increase the development and dissemination of such studies. One of the aims of this study is to increase the awareness of SDA usage through integration with GIS. In the past, solely specific requirements led the researchers to develop new tools. There are several limited methods applied to integrate several models into GIS, like environmental models using tight coupling strategy (Tao, Kainz & Zuidam, 1996). Since urban planning is another part of environmental problems on which GIS experts have focused, there are certain efforts to combine various urban planning models into GIS software, which involves SDA techniques (Hégron, Bocher & Petit, 2012).

Many GIS software packages involve various spatial data analysis tools, such as kernel density estimation, variograms, and kriging (Zhang & Griffith, 1997). However, almost all current commercial GIS packages are limited in terms of providing analysts with broad SDA capabilities. Even ArcGIS by the Environmental Systems Research Institute (ESRI), which has one of the largest user groups among the GIS community, has a wide range of SDA tools, but it has not reached a level at which all significance assessment tests can be performed for SDA. There is an increasing demand for the use of SDA tools in GIS software because such tools provide users with better inference and information extraction capabilities from large volumes of spatial data. For this reason, whenever an SDA tool that is not included in the GIS software is required, users usually convert and import the data to another software package that has the required capabilities and then reconvert the obtained results and import them back to the GIS software for further GIS analyses. This method is costly because it creates several problems. First of all, the geospatial datasets are large and complex in structure and thus cannot be handled appropriately by the software with the required SDA functions. In addition, the topological complexity of spatial data may not be preserved during the importing processes, which increases the workload of the user to establish

the topology again (Griffith, 1993). Moreover, the use of specific SDA software is not always user-friendly because such programs usually require a sophisticated background on the topic. Therefore, the lack of SDA functions in GIS and the need for sophisticated knowledge for interpretation of the abstract and complex nature of the spatial data prevent GIS users from exploring data more deeply and gaining more insight into packages (Zhang & Griffith, 1997). Hence, inference and information extraction from the geospatial data mainly require experienced users in both GIS and SDA and skills in script development. Owing to the rise of SDA and hence the growing GIS user community over the last decade, there is an urgent need for a user-friendly GIS software package with enhanced SDA capabilities.
There are different purposes to integrate SDA and GIS by the developers and the researchers. The sole aim of this integration is not data assessment and sampling. Beyond that, data analysis and exploration should be included in the integration of SDA and GIS.

The need for a properly integrated SDA corresponds to a user-friendly, fast, and visual instrument, which can also be used by non-GIS users and which depends less on other software types, such as statistical packages (Anselin, Syabri & Kho, 2006). The visualization part of GIS provides a strong tool for SDA techniques especially for non-GIS users. The visualization in GIS provides a strong and valuable advantage for SDA packages in understanding and evaluating the analysis of geospatial data. For instance, the Exploratory Data Analysis (EDA) mostly uses dynamic linking and brushing for data visualization by using many different dynamic graphing techniques along with linked scatter plot brushing (Anselin, Syabri & Kho, 2006). In addition, visual information processing is a fundamental approach to non-routine and ill-structured problems (Torun & Duzgun, 2008). Indeed, adding more visual features to SDA tools are the first objective of the integration asbetter visual abilities increase the exploration and evaluation of the data.

When the integration started at the beginning of the 1990s, it was inadequate because various incoherent requirements were claimed from diverse areas like biology, geography, statistics, remote sensing, computer sciences and mathematics. This let developers introcude discipline-specific SDA software packages. Goodchild, Haining and Wise (1992) emphasized that most of the techniques of SDA involved complex and difficult functions, which needed a different approach than the conventional approach of GIS. Since then, various methods have been introduced to achieve the integration of SDA and GIS. For example, Kepoglu (2011) introduced a plugin for full SDA integration in QGIS, which is one of the most widely used open-source GIS software packages. Although this integration had a growing user community, it required updates to be compatible with new QGIS releases (Kepoglu, 2011). Another example is the Info-Map software, which is a standalone and generic SDA software developed in 1995 by Bailey and Gatrell to analyze spatial data quickly (Kepoglu, 2011). The Info-Map contains a wide variety of SDA tools but does not have full GIS functionalities (Mikelbank, 2010). The Info-Map only runs on the Microsoft DOS OS, which is one of biggest disadvantages of the software, and is presently not widely uesed. This shows that the sustainability of the SDA software and GIS and SDA integration is crucial for wider and longer use.

There are various types of approaches to integrate SDA with GIS software like plug-ins and extensions. An extension can be added to the software whether it is closed- or an open-source. If it is closed-source software, then it is called closed coupled software and extensions can add some other features to the software.

There are also extension types in the market. SpaceStat is an extension type of SDA software that can work with ESRI ArcView GIS. Spatial data are transferred from the ArcView GIS version 3 to SpaceStat for analysis, and the result is sent from SpaceStat to ArcView GIS for visualization (Anselin & Bao, 1997).

This type of extension of SDA software can add new menus and submenus to the extended software. In this way, one can perform an analysis using the interface of this extension and visualize the result on the GIS software.

DynESDA is another extension type of SDA tool for working with ESRI ArcView GIS version 3. It was developed according to the idea of dynamic graphics used in exploratory data analysis (Anselin, 2000). New ESRI product (ArcGIS) was started to develop in 1999. Hence the version of ArcView 3 was stopped in 2002. Therefore, neither extension DynESDA nor SpaceStat worked with the new ESRI products and are outdated.

Proprietary software development is another option that has been especially popular among the private sector for commercialization purposes. The development of ArcGIS began in 1999, and it is one of the most widely used commercial GIS software packages in the world. It contains both geostatistics and geospatial tools. The percentage of SDA tools among all tools is only 1.05% in version 9.1 and 1.01% in version 10.0 (Kepoglu, 2011). Although ArcGIS is very powerful software and is widely used across the world, it has two main disadvantages: the first is that it is closed source software, and the second is that it does not have adequate number of SDA tools to fulfill the requirements of the users.

Another important free SDA tool is the Geographic Resources Analysis Support System (GRASS) (GRASS GIS, 2008a). The software is generally used for 'data management, image processing, graphics production, spatial modeling, and visualization' of a variety of spatial data formats (GRASS GIS, 2008a). It supports a wide range of applications and is used in both academic and business worlds (GRASS, 2008a). The project is managed by the international GRASS Development Team (GRASS GIS, 2008a). The software was published as freeware under GNU GPL in 1999 (Neteler & Mitasova, 2007). GRASS runs on UNIX and Windows OS. However, it is not easy to use it on Windows OS without an emulator. It creates difficulties for the user and moreover, GUI is not practical for GIS users.

It is clear that SDA techniques and GIS software complete each other asuccessful and sustainable integration is essential.

Using GIS and SDA, there is no need for investing much, while the development process is continuing. On the contrary, there will be a higher need in investing much for the development of specific integration with different purposes, such as Geoda. One of the most widely known integration software programs, GeoDa was financed by a grant of $4.335.573 (NSF 2007), which was initialized in 1999 and was in progress until September, 2004 (Kepoglu, 2011). There have been various initiatives for different purposes in time to integrate different modules into the GIS software. For different purposes, in order to use spatial analytical functions into GIS and finally use this integration for social sciences, the US-based Center for Spatially Integrated Social Science (CSISS) was established in 1999 as a research infrastructure project funded by the U.S. National Science Foundation (Goodchild, Haining & Wise, 2000).

However, the dissemination has always been a kind of problem for developers to be evaluated and discussed carefully. GeoDais a menu-driven software, which only uses the R techniques to take the input data and show the results on mapview. The user can use it through a point and click interface and does not require any programming skills (Anselin, Syabri, & Kho, 2006). For different purposes, such as for SHAZAM and GIVE, the development and integration are more specific in terms of usage and requirements of users.

For special usage of SDA, there is a great advantage in combining good regression packages and GIS. If GIS is to be developed, convenient computational applications should also be developed, which are compatible with spatial econometrics models (Goodchild, Haining & Wise, 1992). SPACESTAT is an example of this, a SDA package with both exploratory data analysis and regression capabilities. It is coded and compiled in GAUSS, which has a test version. Moreover, Griffith (1993) produced a similar macro in MINITAB, which is shown as a workshop (Goodchild, Haining & Wise, 1992). Even if there were different integrations in the past for specific requirements, the need for the integration of SDA with GIS was emphasized because of the limited number of integration for usage (Anselin, 2000). There are some other modern toolboxes to provide some dynamic interaction with SDA embedded into GIS. This interaction and linkage is achieved by the mapview of GIS software most of the time, such as the linked frameworks combining XGobi or XploRe with ArcView (Cook, Symanzik, Majure & Cressie, 1997; Symanzik et al., 2000); the SAGE toolbox, which uses ArcInfo (Wise, Haining, & Ma 2001); and the DynESDA extension for ArcView (Anselin, 2000), the immediate predecessor programming of GeoDa (Anselin, Syabri, & Kho, 2006). There are several local and small-scale research studies for integrating specific SDA techniques into GIS, such as environmental models, hydrological models, natural disaster models and urban models. The OrbitGIS integration is one example to the integration of urban models into the GIS software (Hégron, Bocher & Petit, 2012). In addition to urban model integration into GIS, there are several other models integrated into GIS, such as the MCDA. It creates and identifies several zones which realize agro-environmental measures, for territory maintenance and for reducing hydro-geological risk (Massei et al., 2012).

If the integration of SDA with GIS is successfully performed, then one can try to add more techniques to analyze the geospatial data with some more advanced SDA tools. Spatial and attribute data are two important elements of the geospatial data for GIS and SDA. There are primarily two statistical techniques dealing with these two elements. The first technique particularly deals with the distribution of spatial data. Furthermore, the technique is applied in point data and sometimes concerns the distribution of point in space whether it is continuous or non-continuous. The second technique is related to the spatial variation of data. This technique is also applied in continuous and non-continuous data, whether the data is point or area type. The exploratory data analaysis improves the capabilities of GIS (Anselin & Getis 1992; Palmer, Bailey & Gatrell, 1996). All the recent attempts aim to increase the overall analytical capabilities of GIS. The geographical information science (GIScience) progressed well due to the improvements in GIS and SDA (Duckham, Goodchild & Worboys, 2003; Goodchild, 1992). In the same sense, by assembling and analyzing diverse spatial data, GIS could have evolved properly (Star & Estes, 1991). All these developments in the study of spatial data and attribute data led to a better understanding of the requirements of GIS and SDA. These requirements are introduced by GIS and SDA experts. GIS experts need more SDA tools and SDA users require GIS capabilities to increase the understanding of spatial analysis and techniques.

Although the main level of the integration of SDA and GIS is created, there has been no standardization about this issue so far (Cremers et al., 1995). The standardization of a framework or methodology has been given importance in recent years, because it creates too many problems for developers especially due to the sustainability and continuity of the development of both SDA and GIS. Creating a standard for data definition, data exchange and concept of Spatial Data Infrastructure provides an interconnection of different systems and a systemic approach (Bocher & Petit, 2012).

In summary, there should be a linkage between GIS and SDA to create and improve more statistical tools for the exploration of spatial data and to analyze the spatial relationships to gather more and meaningful information.

CHAPTER 3

INTEGRATION OF SDA WITH GIS SOFTWARE

In the past, there were various efforts to integrate GIS and SDA using various approaches. These approaches include different components. Coupling strategy is one of the components of the proposed framework. Therefore, coupling strategies are summarized in this chapter. In addition, the SDA software and their examples are defined and explained in greater detail.

3.1. The Coupling Strategies

There are various classifications for coupling any software with GIS. For example, Karimi and Houston (1996) proposed four types of integration methods—namely, isolated, loose, tight, and integrated. Brandmeyer and Karimi (2000) in contrast, identified the integration methodologies in five layers (Figure 2).

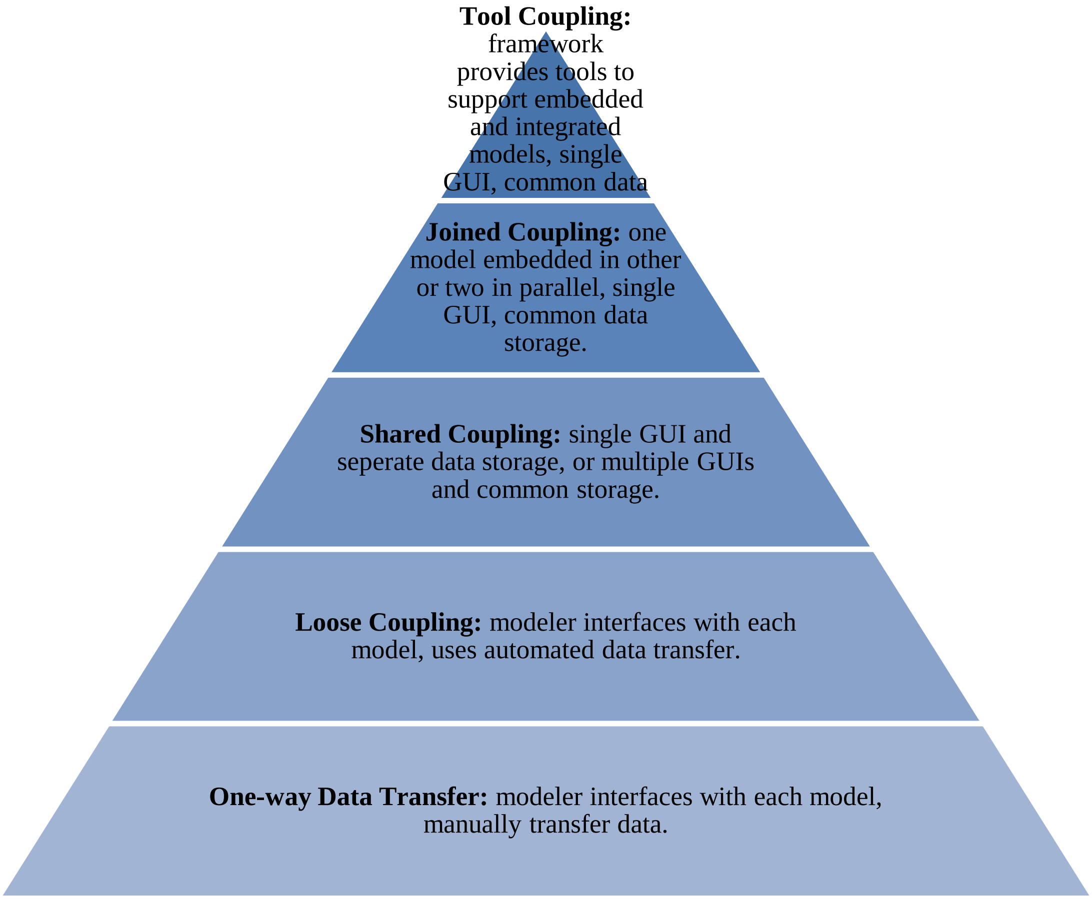


Figure 2. Progression of Coupling Strategies (adapted from Brandmeyer & Karimi, 2000)

The functionality, interface, integration and interoperability of these systems are critical parameters for integrating different systems (Sibolla, 2009; Cascadoss, 2007). This implies that the data and other shared objects should be used by two coupled systems without any problems, and they also should be independent of each other. Independency was defined by Nyerges (1992) as GIS and SDA techniques performing different purposes without any interaction (Brandmeyer & Karimi, 2000). Various researchers proposed models for integration of different software programs (Karimi and Houston, 1996; Brandmeyer and Karimi, 2000; Mitasova and Mitas, 2004 as quoted by Yassemi et al., 2008; Sui and Maggio, 1999 as cited in Brimicombe, 2003).

The types of coupling strategies between SDA and GIS are summarized as loose, thight and embedded coupling in Figure 3. As these are very similar to the bridges between spatial modelling software and GIS (Batty & Xie, 1994). The coupling strategies depend on the customization level for vendors. However this is not the only criteria for the selection of integration methodologies, and one should be aware of the factors of time, users, performance, flexibility, OS and several others before development.

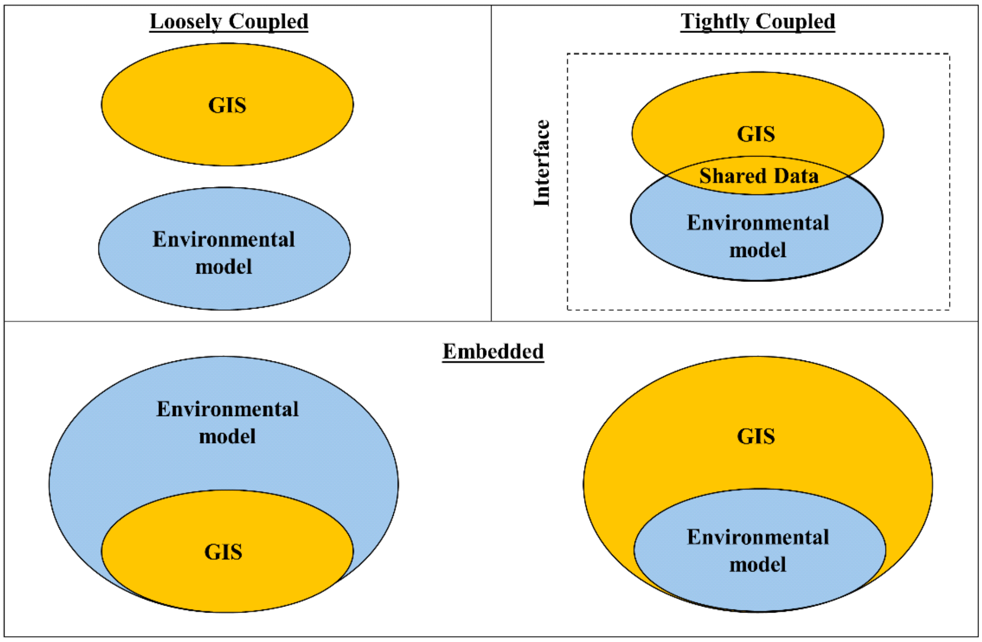


Figure 3. Levels of Integration between GIS and Environmental Models (adapted from Brimicombe, 2003)

3.1.1. The Loose Coupling Strategy

The loose (close) coupling involves exporting and importing data between the two systems (Brimicombe, 2003), and GIS also visualizes the output of SDA techniques. Effectiveness is a critical parameter for coupling, but loose coupling is the slowest among the coupling strategies; therefore, it is not generally preferable. It is also accepted as the lowest level of coupling strategy (Karimi & Houston, 1996). The conversion requirements of the data hinder the link between SDA and GIS software (Fotheringham & Rogerson, 1994) as users certainly loose the originality of the data while making the conversion. To solve this problem, ASCII files is commonly used for conversion purposes. However, this problem also requires back and forward conversion to adopt the data for the SDA and GIS software. Moreover, input and output requirements create further problems.

3.1.2. The Tight Coupling Strategy

Tight coupling strategy encompasses data and graphical user interface (GUI) integration, as described by Brandmeyer and Karimi (2000). Tight coupling is the case in which GIS and SDA techniques can be accessed via a common interface and usually models shared data files. If the two systems require different data formats, the conversion is captured in the integration without interference by the user. The interaction of two systems is achieved by a common GUI. It aims to embed SDA into GIS software (Fotheringham & Rogerson, 1994), which implies that the addition of a new main menu, sub-menus and toolbox requires the GUI of GIS software. Users utilize this menu/submenus/toolbox to initiate the interfaces for each SDA technique. Input/output and other constraints can be defined on these interfaces. Tight integration is a better strategy than loose coupling, because it involves calling SDA routine from the GIS (Bailey & Gatrell, 1995). Most importantly, if there is no macro language embedded into the GIS software, it is the only and the best solution for a successful integration. That is why this integration strategy is implemented in this study.

3.1.3. The Embedded (Full) Coupling Strategy

Embedded (full) coupling strategy refers to cases in which either SDA techniques are built in GIS or GIS functionality is available in SDA (Sui and Maggio, 1999). At the beginning of the 1960s, programmers embedded the functionality of SDA or other techniques into software by using macro languages or scripts. An example of embedded coupling is a GIS system that has a programmed environmental model executing within the GIS (Brimicombe, 2003). The embedded method is better than the tight integration methodology because of the fast data transfer capabilities and higher level of integration. However, it requires more programming skills, time, and a larger project group for a successful integration. It is even not easy to find an embedded macro language open-source GIS software to develop more tools on it.

3.2. Overview of Existing GIS and SDA Software

SDA and GIS experts and software engineers have developed various software programs using various techniques, methods and programming languages. These programs have been used for many years by users for various purposes. However, most of them are outdated and currently out of use and are not developed any more. They are summarized in Table 2.

Table 2. SDA Software Examples and their Usage (Kepoglu, 2011)

Type of Software	**Software**	**Advantages**	**Disadvantages**	**Status**
Stand-alone SDA Software	Info-Map	software package distributed with a book for interactive SDA	a closed-source package and DOS-based product	quite outdated; antiquated architecture and performance constraints
	SpaceStat	Provides tools for creation of spatial weights matrices, exploratory SDA, spatial econometric analyses.	a closed-source package and Windows-based product	quite outdated; antiquated architecture and performance constraints
	GeoDa	Free-user friendly software, supporting interactive EDA	a closed-source package and Windows-based product	running only on Windows XP OS, quite outdated
	CrimeStat	spatial statistics program used in crime mapping, links with most of desktop GIS software like ArcGIS and MapInfo	a closed-source package and Windows-based product, no display	running on recent OSs

Table 2. (Continued)

Close Coupled SDA Extension	SpaceStat Extension	Provides tools for exploratory SDA, an extension program for ArcView GIS series 3	a closed-source package and close coupling	quite outdated; antiquated architecture and performance constraints
	DynESDA Extension	Dynamic exploratory SDA, an extension program for ArcView GIS series 3, brushing and linking maps	a closed-source package and close coupling	quite outdated; antiquated architecture and performance constraints
	sd4pp Extension			
Proprietary Desktop GIS Software	ArcGIS	Advanced Desktop GIS program, full rich of GIS tools, worldwide used	a closed-source package and very expensive product	running on recent OSs
FLOSS Product	R	An FLOS statistical package, have many tools in variety of fields including SDA, cross platform support	not a user-friendly environment, requires programming	running on recent OSs
	GRASS GIS	FLOS GIS software, cross platform support	a few SDA tools, Unix style usage, different data format	running on recent OSs

In reality, there are many SDA and GIS tools in the market; however, the main problem about the current software types is that they all have different requirements and features.

3.2.1. Isolated (Stand-alone) SDA Software

A typical stand-alone and generic SDA software example is Info-Map, which was developed in 1995 by Bailey and Gatrell to analyze spatial data (Kepoglu, 2011). Info-Map contains a wide variety of SDA tools and it is not entirely a GIS program. Info-Map has some simple mapping facilities for the exploration of data and the display of the results of the analysis (Mikelbank, 2010). Info-Map performs the following SDA tools:

- Kernel Estimation,
- K Functions,
- Variogram Estimation,
- Kriging,
- Spatial Autocorrelation Analysis,
- Spatial Regression,
- Principal Components Analysis,
- Multivariate Classification, and
- Non-Spatial Regression.

Info-Map only runs on the Microsoft DOS OS, which is one of the most important disadvantages of the software and is currently not widely uesd. Another disadvantage of Info-Map is that the software was developed in a closed source environment.

Another example is SpaceStat, which was developed by Anselin between 1983 and 1994. Anselin (1992) explains that SpaceStat was first released in 1992 by the US National Center for Geographic Information and Analaysis (Kepoglu, 2011). The main disadvantage of this software is the OS on which it runs. It only runs on 32-bit DOS environment and on the Windows OS. SpaceStat can work as stand-alone software and as an extension. It requires ESRI ArcViewseries 3. This is another disadvantage of this study. In order to overcome such problems, SDA with GIS is to be run in all the versions of the selected GIS. Like Info-Map, SpaceStat is also a closed-source software. The snapshot of the SpaceStat is shown in Figure 4.

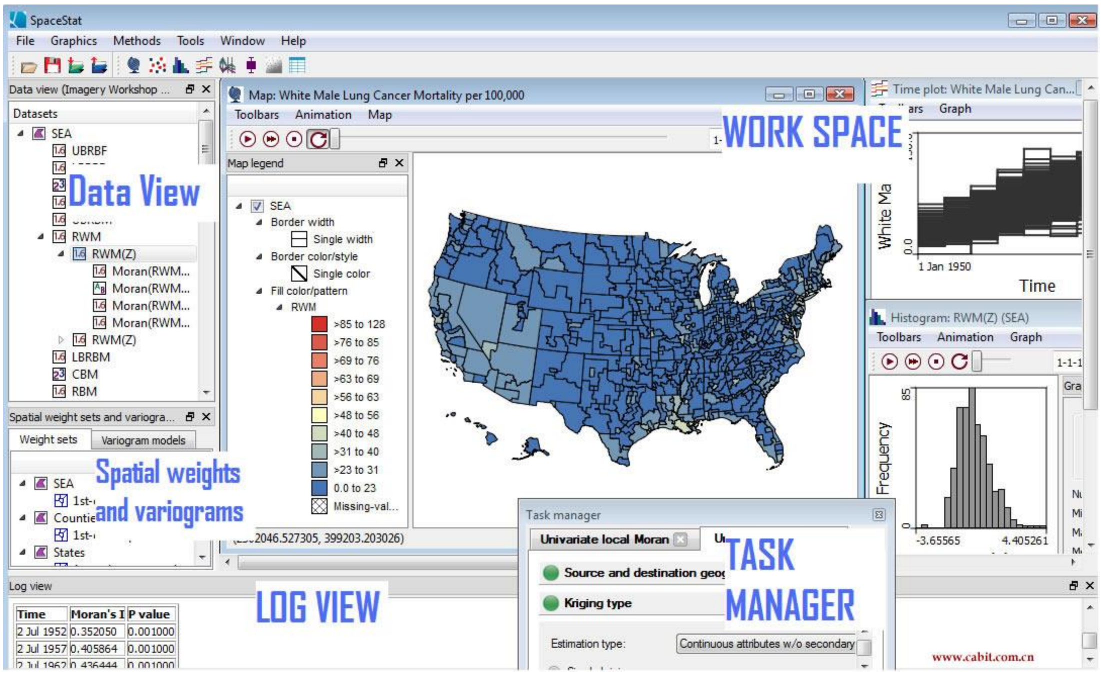


Figure 4. SpaceStat Interface and Analysis Tools

GeoDa is another stand-alone software. It has a user-friendly interface as shown in Figure 5. This is one of the most advantageous aspects of GeoDa compared to the other software types. In this way, one can use the software without any knowledge of GIS. It runs on the Windows OS and is not open-source. It is defined that GeoDa is a reinvention of the SpaceStat (Anselin, Syabri & Kho, 2006). Even if it is the reinvention of SpaceStat, there is no different methodology used for GeoDa other than SpaceStat. Therefore, it is inevitable for GeoDa to share the same fate with SpaceStat. The OS of GeoDa is also Windows. Such SDA software working only on the Microsoft closed-source environment has become quite outdated. In 2007, the GeoDa team realized that GeoDa did not work on Windows Vista and after that a new study had to be initiated to provide software for all platforms. After that, GeoDa began to be used on Windows (XP, Vista and 7), Linux (with both 32- and 64-bit) and Mac OS X (Intel and PPC). The new development is totally different from GeoDa and therefore called OpenGeoDa (Figure 6).

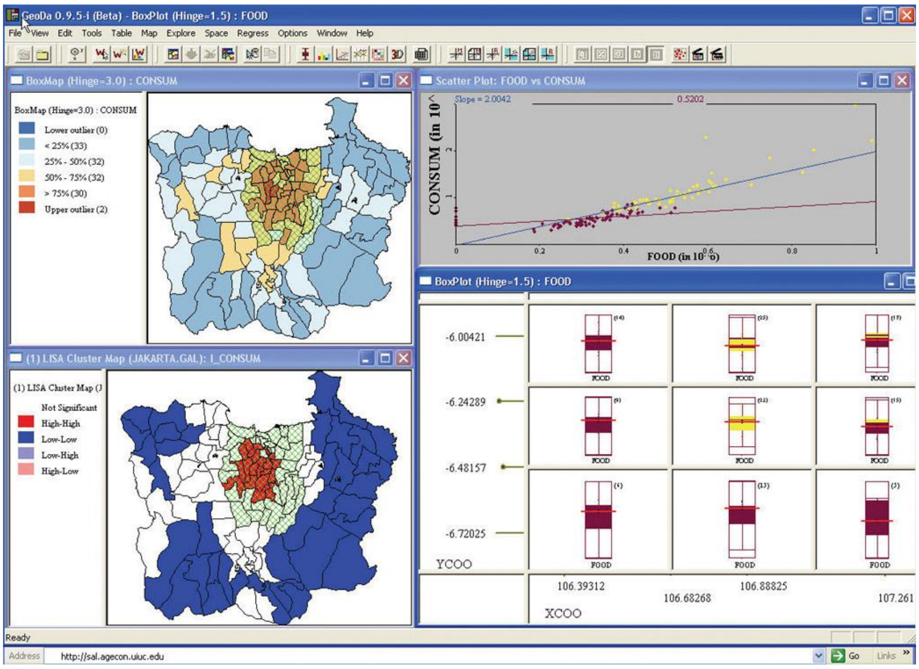


Figure 5. Snapshot of GeoDa Version 0.9.5-I (Beta)

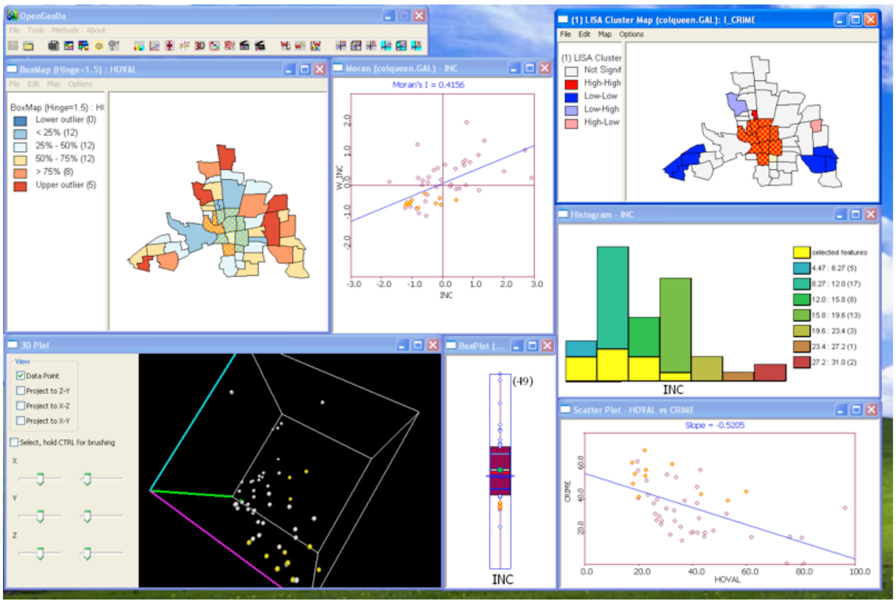


Figure 6. Snapshot of OpenGeoDa

CrimeStat is another stand-alone SDA software (Figure **7**). The software is used for crime analysis and mapping purposes. It has point pattern analysis capabilities. The software can read “DBF”, “SHP”, ASCII or ODBC-compliant formats (Levine, 2006). Having different types of data input is one advantages of this software; it proves that the integration of SDA and GIS should accept different types of input for analysis purposes. This will provide a strong tool and different choices for end users. CrimeStat has not included any visualization environment, which makes it useless for end users. Moreover, the source code of CrimeStat is closed and it only works on the Windows OS.

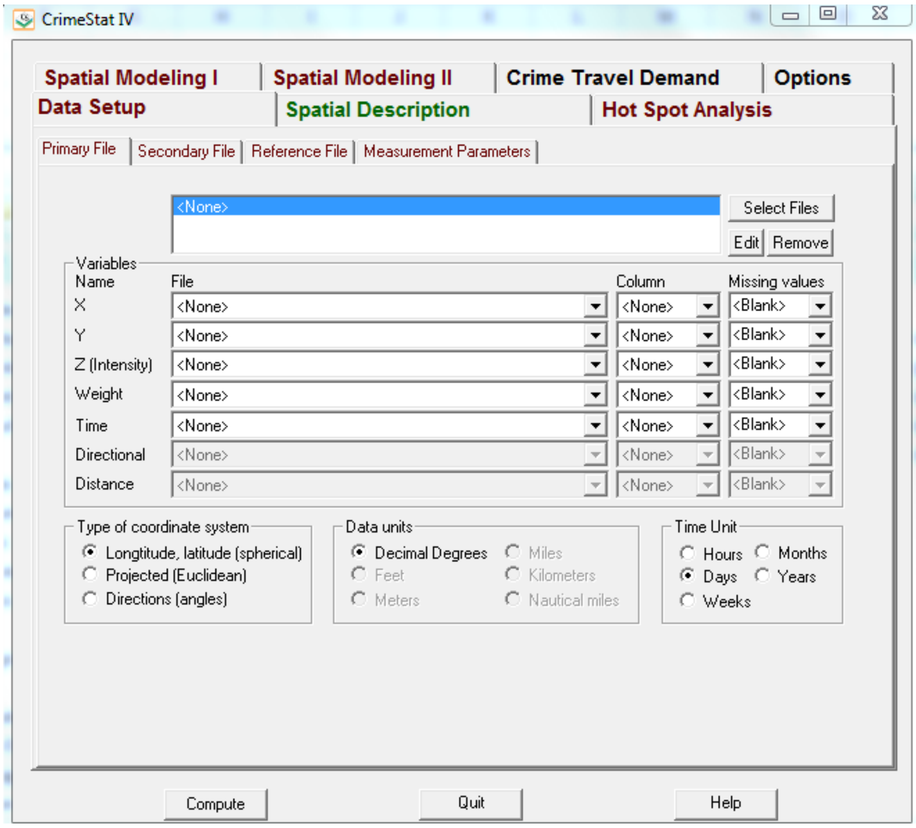


Figure 7. Snapshot of CrimeStat

Other than the above mentioned stand-alone SDA software, there are some other stand-alone freeware software available for analyzing geostatistical data as shown in Table 3.

Table 3. Stand-alone Free SDA Software (Ai-geostats.org, 2015)

Number	Software Name	OSs	Development Languages	SDA Techniques
1	Agromet	Unix Windows	C++	V-K-C-2D *
2	Cosim	Windows	Fortran	S-2D
3	ExploStat	Windows	-	V-K-O-C-2D-G
4	E{Z}-Kriging	Windows	-	V-K-2D
5	GCOSIM3D	Windows	C	S-3D
6	Geo-EAS	Windows	-	V-K-2D
7	GeoPack	Windows	-	V-K-C-2D
8	GeoXP 1.3	Windows	-	V
9	Geostatistical Toolbox	DOS	-	V-K-C-3D
10	GMT	Unix	C	O-2D
11	GRNN	Windows	-	O-2D
12	GSLIB	-	Fortran 77	V-K-C-3D-S
13	GStat	Windows Linux	C R	V-K-C-3D-S
14	ISIM3D	Windows	C	S-3D
15	IV	Windows	-	V-2D
16	Kriging	Unix	C	K-2D
17	SADA	Windows	-	V-K-O-3D-G
18	SAGA GIS	Windows Linux	-	V-K-2D-G
19	SGS	Linux	C	K-S-2D
20	S-GeMS	Windows Linux	C++	V-K-3D-S
21	SpatDesign	Windows Linux	Matlab Octave	V-K-O-3D
22	Spherekit	Unix	C	K-O-2D
23	Surface III	Mac	-	K-O-2D
24	Surfit	Windows	C++	O
25	UNCERT	Unix	C	V-K-O-C-2D-G
26	Variowin	Windows	-	V-2D
27	Vesper	Windows	-	V-K-2D

* (Keywords: V = Variography, K = kriging, O = other estimators (NN, IDW, splines...), C = co-kriging, 2D/3D. = max dimensions, S = simulations, G = GIS functions).

In addition to free stand-alone SDA software, there are some commercial SDA packages (Table 4)

Table 4. Stand-alone Commercial SDA Software (Ai-geostats.org, 2015)

Numbers	**SDA Software**
1	Datamine
2	Geosotokos Toolkit and ECOSSE
3	Geovariances' ISATIS
4	Geostat Systems Int. Inc.
5	GS+ from Gamma Design
6	Surfer from Golden Software
7	SAGE2001
8	Lynx Geosystems
9	GeostatsOffice
10	SurGe
11	EGS-Enjoy Geostatistics
12	Auto-Multi-Linear-Fitting

3.2.2. Closed Coupled Software Including SDA Toolbox

Extensions can be appended to the software after setup. The software can be set up and can work without an extension, but extensions can add certain other features to the software. An extension can be added to the software whether it is closed- or open-source. SpaceStat (Figure 8) is an extension type of SDA software that can work with ESRI ArcView GIS. Integrated techniques are mainly composed of exploratory SDA tools (Anselin, 2000). Spatial data is transferred from ArcView GIS series 3 to SpaceStat for analysis, and the result is sent from SpaceStat to ArcView GIS series 3 for visualization (Anselin & Bao 1997). This type of extension of SDA software can add new menus and submenus to the extended software. In this way, an analysis can be performed using the interface of this extension and the result can be visualized on the GIS software.

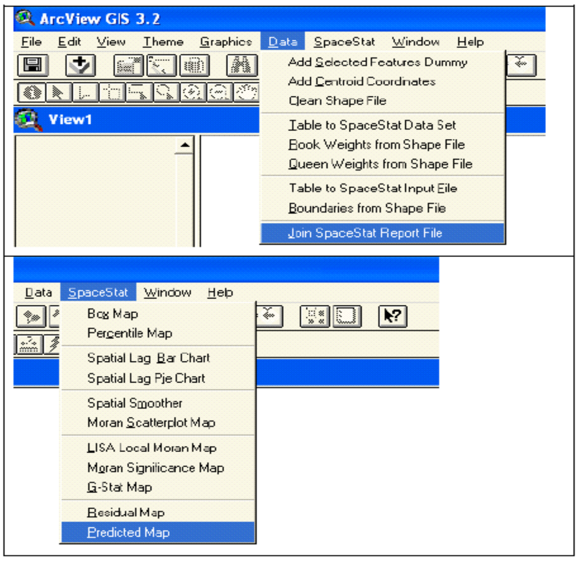


Figure 8. Menu Items of SpaceStat Extension in ArcView GIS Series 3

DynESDA (Figure 9) is another extension type of SDA tool working with ESRI ArcView. The DynESDA extension for ArcView GIS series 3 was created with the design of dynamic graphics as in EDA (Anselin, 2000). ESRI started to develop the code for ArcGIS in 1999. The development of ArcView GIS series 3 was terminated in 2002 (Kepoglu, 2011). Therefore, both DynESDA and SpaceStat extensions did not work with the new ESRI products and are outdated.

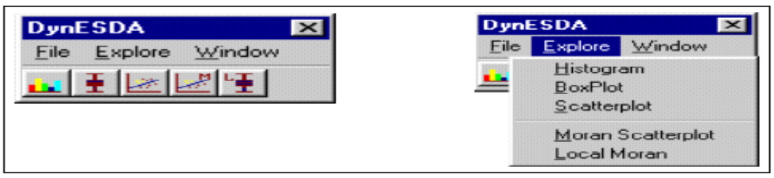


Figure 9. DynESDA Extension in ArcView GIS Series 3

3.2.3. Proprietary Desktop GIS Software Including SDA Toolbox

ArcGIS began to be developed in 1999 (Figure 10). ArcGIS Desktop is one of the most widely used commercial GIS software in the world. It contains both geostatistics and geospatial tools. Though ArcGIS is a very strong software program and is widely used throughout the world, it has two main drawbacks: being a closed-source software product and not having wide range of SDA tools to fulfill the user requirements. These two drawbacks prove that there should be new development tools working with an open-source software package and there is a need for more tools to fulfill the requirements. Other proprietary software like MapInfo, Intergraph has similar drwabacks.

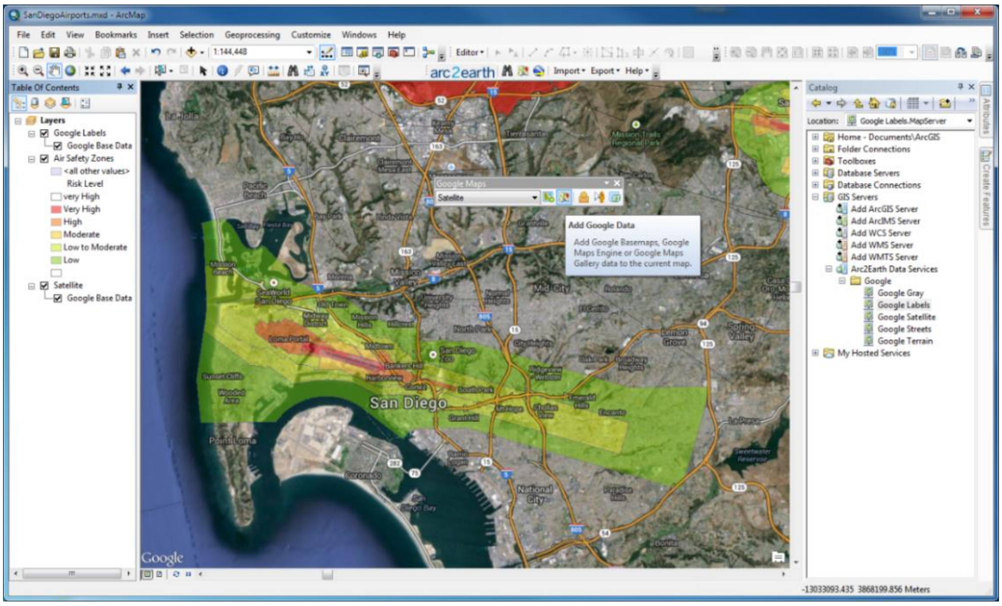


Figure 10. Snapshot of ArcInfo Licensed ArcGIS Desktop Version 10.3

3.2.4. FLOSS SDA Software

There are some tools in the market used without GIS integration. However, Graphical User Interfaces (GUI) of these tools are not mostly user friendly. One of the most widely used one is R (Figure 11). This tool is not only used for SDA purposes, but also for image processing and statistical purposes. According to Venables et al. (2008), the usage of R is mostly for "data manipulation, calculation, and visualization". R is more for mathematical calculations and displaying graphics (Maindonald & Braun, 2006). The origin of R started with a research project in the 1990s and it has been a GNU project since 1997 (Chambers, 2008). R is a syntax-based language. There are many different packages for different purposes. These packages are developed by different third parties. Many statistical techniques are supplied by packages (Venables et al., 2008). Thousand packages in the central repository of the R project support the base system (Chambers 2008). Each package can be downloaded and added to the R software and then scripts can be used. One advantage of R is that it is a Free and Open-Source Software (FOSS) program. However, it has certain disadvantages, though. R is not user-friendly because it is dependent on script. This feature makes the use of R rather difficult. Therefore, learning each package is a difficult and a time-consuming process. It requires experience about statistics, programming languages and mathematics. The use of R is not easy particularly for GIS users who are accustomed to working with GIS. R can run on all OS's, which makes it attractive for integration as well.

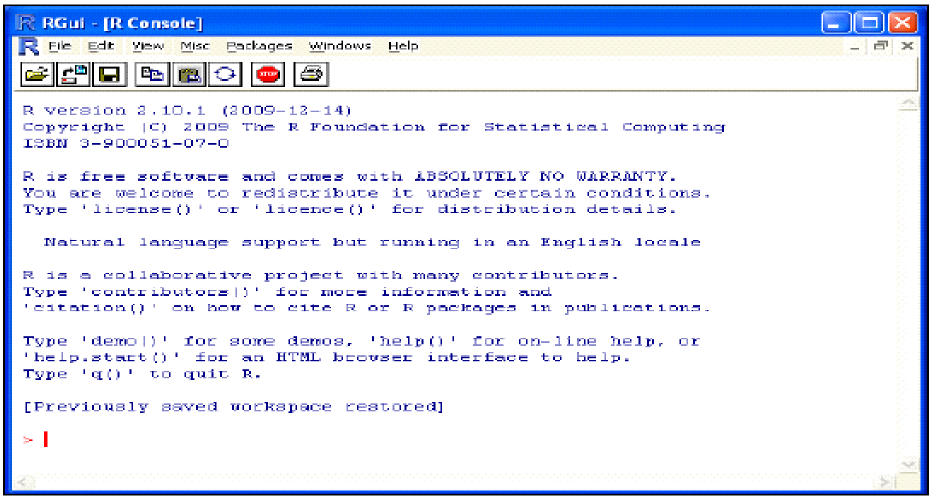


Figure 11. Snapshot of R GUI

Another important free SDA tool is the GRASS (Figure 12). GRASS is usually used for data management, graphic processing, visual production, and spatial modeling (GRASS GIS 2008a). GRASS alters raster and vector data, changes multispectral visual data and saves spatial data (GRASS GIS 2000b). It can be accessed by a user interface or a command prompt, providing support for many applications that are used by the industry and academicians (GRASS GIS 2008a). The system was produced by the GRASS development team in 1999 and is free under GNU/GPL (Neteler et al., 2008). GRASS works on UNIX and Windows OS’s. However, it is not easy to use it on Windows without an emulator. This makes it difficult for the users. Besides, GUI is not usable for GIS users.

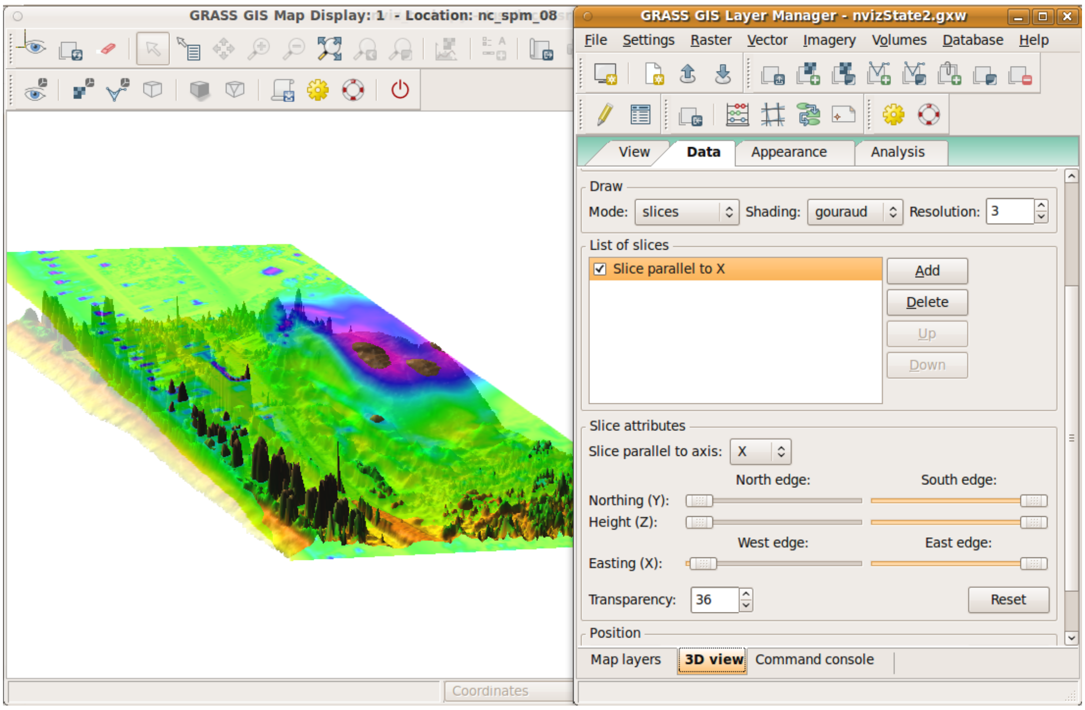


Figure 12. Snapshot of GRASS GIS in Ubuntu OS

The Free Software Foundation defines the free software as: "Free software is a matter of the users' freedom to run, copy, distribute study, change and improve the software. To be clearer, it should provide four freedoms for developers and users:

- Freedom 0: Developers/users can run the program for any purpose (freedom 0).
- Freedom 1: Developers/users can study how the program works, and adapt it to their development. It means that they can access the source code of the program.
- Freedom 3: Developers/users can redistribute copies.
- Freedom 4: Developers/users can improve the program, and release their improvements to the public, so that the whole community benefits (Akbari & Rajabi, 2013).

The reasons mentioned above for the users and developers are really crucial for the sustainability and the dissemination of the software in particular. Therefore, the open-source development strategy is applied in this study not only to select the software which has been integrated but also to select the bridge technique. As demonstrated in this study, the open-source paradigm perfectly addresses the needs and the development methodology of an integrated modelling platform dedicated to sustainable software development.

Open-source software and practices can have major empowering impacts on software development and distribution. The free availability of the software also offers a number of advantages for the academic world. In this way, students, researchers and academicians can set up the software on their personal computers in order to use the development. This brings about a freedom for users to use or develop the software by exploration and discovery by themselves and at their own pace (Rey, 2012). The open-source strategy also overcomes the problem of "black box" methodology, which hinders the implementation of a method or algorithm, and it brings a really preferable advantage to the open-source strategy.

Recently, the usage of FOSS development has been on the increase, especially in the field of Geospatial Information Systems (Akbari & Rajabi, 2013). Since the usage of open-source software brings about financial benefits to the government, businesses world, and non-profit organizations, this type of software has been preferred by these organizations (Confino & Laplante, 2010). In this way, students, academicians, governmental organizations and private organizations which have a low capital can use it without any investment. This increases the dissemination and usage of study as well. Obviously, the ongoing awareness of FOSS tools in GIS community helps with further expansion of these tools to new applications and solving other problems. Furthermore, software and algorithms developed in research projects are increasingly being published under open-source licenses (Akbari & Rajabi, 2013). This approach guarantees sustainability, because researchers can have a chance to use open-source codes to develop the software further. Dissemination and sustainability is also the aim of this study to get more feedback and improvement from the developers and users.

Hence, the open-source GIS and SDA are preferred and applied in this study. The reuse of the source code and merging into different software is the foremost advantage of the FOSS. The selected GIS and SDA software and their selection are explained in more detail in the following chapters. However, the criterion of the source type affects the selection of software. As it was pointed out before, the bridge method is also selected from open-source interpreters. Table 5 summarizes some other advantages and disadvantages of open-source GIS.

Table 5. Advantages and Disadvantages of Open-Source GIS (Akbari & Rajabi, 2013)

Advantages	**Disadvantages**
It allows to change and modify the source code	Code quality and exact testing procedure is not under guarantee
It can be easy for many people to access the source code	Morality of the code is not under guarantee
It can fix bugs immediately	Problems in utilizing some open-source software as a basis for a business
It is free to use, free to modify, free to update and free to distribute	

CHAPTER 4

THE PROPOSED FRAMEWORK

In this chapter, the proposed framework is explained with its steps. These steps are also composed of different sub-steps. Each step with their sub-steps are described.

4.1. The Proposed Framework

The proposed faremwork is based on integration of uDig GIS sotware with R SDA toolbox by using RCaller bridge techniques. The framework has four main steps (Figure 13.). The first step is requirement analysis, which is based on components of GIS (i.e. data, software, hardware, and user) and SDA techniques. The second step is determination of integration methodology, in which coupling strategy and bridge technique are identified considering the requirement analysis. The third step is software integration which is basically an architecture definition, where programming language(s), development environment and implementation of the methodology are selected and implemented. The selected GIS and SDA software should be suitable for the coupling strategy and programming language(s). This step contains intensive script development and coding with the selected programming languages. The final step covers tests, where the integration is tested with alpha (developer) and beta (user) tests. If the developer test is successful, then the user test is applied; otherwise, the developer test is iterated until a successful implementation is achieved.

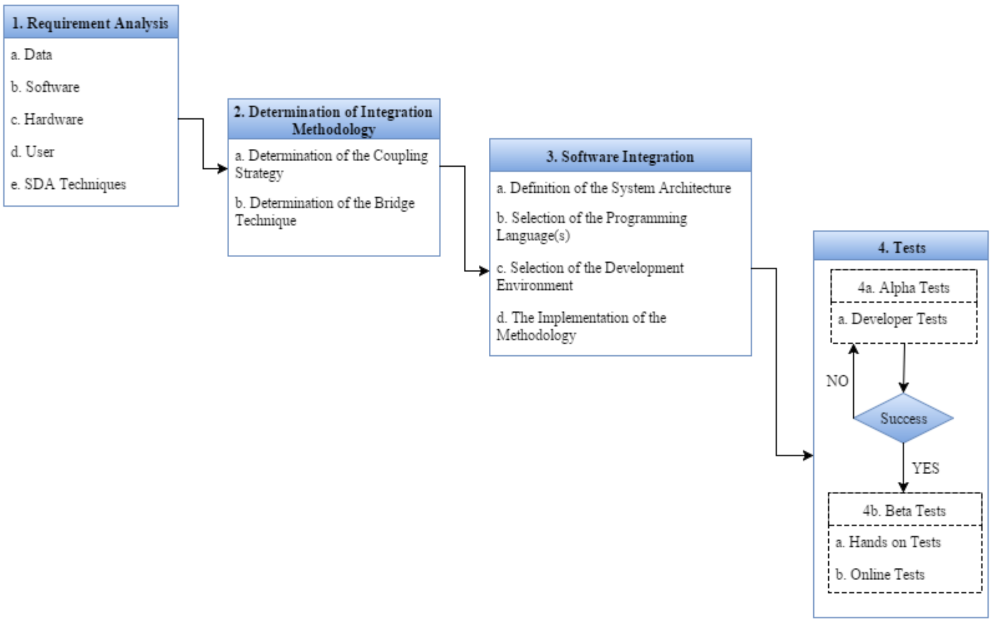


Figure 13. The Proposed Integration Framework of SDA with GIS

4.1.1. Requirement Analysis

Requirement analysis is composed of the analysis of components of GIS and SDA techniques. The requirements related to the each component of GIS (data, software, hardware and user) as well as the SDA techniques are defined:

Data:

Data are the input of the system and must be suitable for both the GIS and SDA software requirements. As explained before, most of the users have serious problems with the conversion of data in currently available software programs. Although the ESRI shapefile format is accepted as a standard for the GIS software, there are various types of data in hand owing to the usage of different hardware and software. The users must use XLS, MDB, CSV and ASCII types of data without conversion. Therefore, a user-friendly GUI is designed and integrated in GIS for the usage of the GIS community.

Software:

The selection of GIS and SDA software should be evaluated with respect to functionality, sustainability, reliability, efficiency and usability. In addition to these criteria, the sustainability of the GIS software should be taken into account, which specifies whether the GIS software is continuously developed or not. If a community supports the project, it means that it can be reliable and sustainable for a while.

Another important point to keep in mind while selecting the GIS software is whether it has an open-source environment or not. It is absolutely essential for this study to select open-source SDA, GIS and Bridge Technique. If the strategy is selected as free and open-source, one should select all the components of the framework from the FOSS choices. Otherwise, it creates irreparable problems for developers during the development process.

Moreover, the architecture of the selected software allows the developers to embed another software or toolbox into it. Since the architecture of different software allows the developers to reach a limited amount of code or development, the architecture of the software should also be suitable to be integrated without too much effort.

The factors of cross platform suitability, 32/64 bit support, extensive documentation and being free-to-distribution should also considered as they increase the usage, distribution and dissemination of the study.

Hardware:

Since hardware is not a key factor for the success of this study, it is not explained in detail, but one should know that hardware is a component of GIS and should be accounted carefully to meet at least the requirements of software and development tools. It means that the setup file must run the software without problems in the computer.

User:

The two different users namely, the users of GIS software and the the SDA users should be equally considered. For this reason, the requirements of users are defined and analyzed in literature survey part (Chapters 2 and 3). Since there are at least two different software types integrated, the user requirements should be defined in a systematic way.

SDA Techniques:

In this step, the methodology and the bridge technique, which are two key factors, are evaluated and decided. IFor this reason, the integration types discussed in Chapter 2 and 3 should be carefully evaluated. The selected integration typesis explained in the following sub-section. In addition, the bridge technique to link any programming language with the R SDA toolbox should be determined by selecting appropriate R interpretter. Therefore, the R interpreter types should also be carefully evaluated and selected.

4.1.2. Determination of the Integration Methodology

In this step, the methodology and the bridge technique, which are two key factors, are evaluated and decided. For this reason, the integration types discussed in Chapter 2 and 3 should be carefully evaluated. The selected integration typesis explained in the following sub-section. In addition, the bridge technique to link any programming language with the R SDA toolbox should be determined by selecting appropriate R interpretter. Therefore, the R interpreter types should also be carefully evaluated and selected.

Determination of the Coupling Strategy:

Developers can use the proposed framework to select different SDA techniques, coupling strategies, interpreters and GIS software to develop their own methodology for a successful integration. For this purpose, the coupling strategy depends on the selected components of the framework and their adaptability. Various classifications for coupling two software packagesare discussed in Chapter 3. Existing coupling strategies discussed in the literature are compiled in this thesis (Table 6). Table 6 also provides pros and cons as well as their main characteristics.

The integration methodologies given in the literature with their pros and cons as well as their main characteristics are listed in Table 6.

Table 6. Summary of Coupling Strategies (compiled from Bhatt, Kumar & Duffy, 2014; Ungerer and Goodchild, 2002; Goodchild, Haining & Wise, 1992; Sui & Maggio, 1999; Brimicombe, 2003)

	Integration Methodologies			
Characteristics	**Isolated**	**Loose**	**Tight (Close)**	**Integrated (Embedded)**
Shared User Interface			X	X
Shared Data and Method			X	X
Intra-simulation Model Modification				X
Intra-simulation Query and Control				X
Description	Analysis and output display directly in spatial analysis software	Analysis in spatial analysis software, output display in GIS, facilitated by online file database exchange	Analysis method varies; GIS and analysis package share a common GUI and data.	Analysis and output display directly within GIS
Advantages		Less work in terms of code creation	-Spatial analysis can be done within the GIS environment -Data exchange is automatic between GIS and SDA Techniques Merges different tools in a single powerful system -Avoids inconsistency and data loss originating from redundancy and heterogeneity of method base	-No file import or export, no code creation required - Steerable simulation in terms of possibility of changes in parameters or processes

Table 6. (Continued)

Disadvantages	-Abundant GIS data layers cannot be used	-Consuming time to import and export data -Information sharing through exchange of files, which can be inefficient and error prone -Different tools and libraries facilitate independent development -Distinct GIS and SDA packages with independent interfaces	-Much work in terms of code creation	-Possible lack of specialist insight in spatial analysis -Significantly complex programming and data management - Changes to the code base are not easy because of its embedded large source code structure
Integration Strategy	Data transfer off-line	Reference/cross index on-line	Interoperable store services	Common store services
Data model similarity	Different or same	Different or same	Different or same	Same
Data model construct resolution	Manuel resolution of data construct differences	Manuel resolution of data construct differences	Automatic resolution of data construct differences	No need for data construct resolution
Data Communications	Not needed	High speed phone or ethernet	Data bus or ethernet	Data bus
User Interface	One-off	Few	Often	All the time
Sped of Interaction	Slow	Tolerable	Reasonable	Fast
Amount of Software Development	None-low	Low-some	Some-lots	Lots
Hardware Platform	Different or same	Different or same	Different or same	Same
Personnel Involved	Different	Different or same	Different or same	Same
References	Nyerges, 1992;	Coroza, 1997; Storck et al., 1998; Hellweger and; Nyerges, 1992;	Lahlou et al., 1998; Liu, 2005;	Huang and Jiang, 2002; Kinerson et al., 1999; Nyerges, 1992;

Developer should select the coupling strategy with respect to the requirements, the adaptability of software with coupling strategy and the pros and cons. The tight coupling strategy adopted in this thesis is shown in Figure 14.

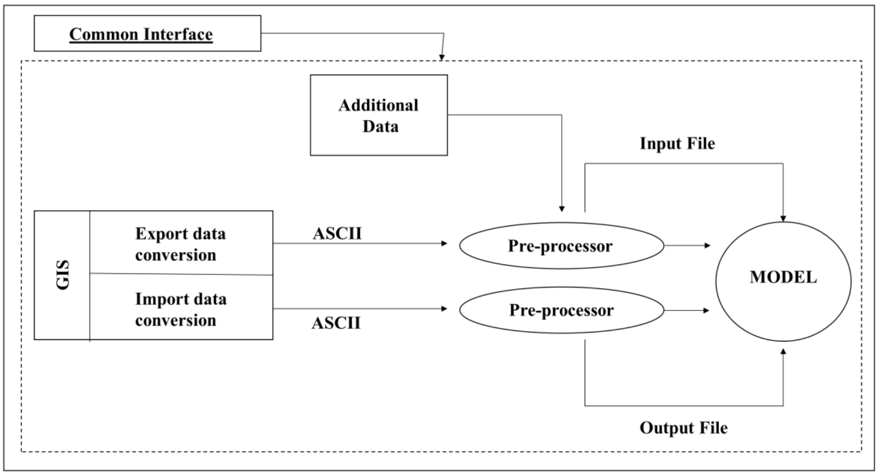


Figure 14. Tight Coupling Strategy

Determination of the Bridge Technique:

The integration can be achieved by using different bridge techniques and programming languagese. The selection of the bridge technique is also dependent of the requirement analysis. As this thesis study aims to provide a FOSS for users and developers, the bridge technique criterion is also free and open-source. Furthermore, the programming language of the bridge technique is suitable and compatible with the SDA and GIS software.

4.1.3. Software Integration

In this step, the programming language(s) and the development environment are selected with respect to the requirement analysis and designed integration methodology.The Software Integration step has four cmponents namely, “Definition of the System Architecture”, “Selection of the Programming Language(s)”, “Selection of the Development Environment” and “Implementation of the Methodology”.

Definition of the System Architecture:

Before implementing the methodology, several key factors affecting the implementation should be taken into account. One of these key factors is to understand the architecture of the GIS software in every detail in order to embed the SDA software without experiencing any problems. Understanding the architecture is to clarify whether the system is suitable to embed selected SDA techniques for integration or not. It is not only important for SDA techniques but also for programming language(s) and development environment and their suitability. Once the architecture of the GIS is understood, it should be restructured with the integration of the SDA software and its techniques.

Selection of the Programming Language(s):

The language for development purposes should be carefully decided during the problem statement stage. It is a very critical decision to select correct programming language(s) for usage, development, dissemination and sustainability. Since there are two different software types, there should be at least two programming languages used in this thesis. The programming language does not affect the success of the study but may affect the performance. Therefore, the performance and sustainability are two key factors for programming language selection. Another factor to be considered is the usage and spread of language to reach related developer community.

Selection of the Development Environment:

The development environment refers to software coding, debugging and testing environment. Although it is not so critical on which environment the development has been carried out, it is selected and created carefully for ease and suitability of development.

The Implementation of the Methodology:

This step is based on the the coupling strategy. Each component of the coupling strategy is the same with the proposed methodology. However, the communication, connection and some components are restructured with respect to the requirements, selected components and performance criteria. In this thesis, a methodology is provided for mid-level developers to follow and apply each step in order to integrate any SDA technique into GIS successfully. This sub-step is the most critical step of framework.

The implementation methodology should follow Software Development Life Cycle (SDLC), which is re-design of one of the existing SDLC methods. A typical SDLC have steps of analysis, design, implementation and testing (Burback, 1998). Hence these four steps are also adopted in this study. The methodology is defined as standardization of each step defined and explained in framework. Henning (2004) defined the methodology as the coherent steps that complement each other. Also, Mouton (1996) describes the methodology as the ways of achieving a specific goal. Finally, the formal definition of methodology is that a set of methods, rules, or ideas that are important in a given discipline (Anon, 2016).

The the implementation methodology developed in this thesis are shown in Figure 15. Figure 15 illustrates the steps of the proposed methodology so that developers canintegrate any R technique into uDig. Spatial data sources for this module iinvolves all data that can be accepted by uDig (the first stage in Figure 15), such as MDB, TXT, CSV, XLS and ArcView shapefiles. Once these data are read by uDig as an input, an import/export wizard i converts the data into a shape file if required (the second stage in Figure 15). The input file is then used by structured RScripts, and spatial data constraints are defined and implemented in the R environment (the third stage in

Figure 15). The result of the R environment is executed in the R environment and displayed outside of GIS (the fourth stage in Figure 15). The RScript is embedded in the fifth stage (Figure 15) using an R interpreter, which is called RCaller. In stage sixth (Figure 15), the input, output and other constraints are defined for the design of the GUI of each technique. In stage 7a (Figure 15), the interface is designed by using the Window Builder to diminish the workload of developers. To finalize the interface design and development, the integrated structured RScript is implemented into a graphical user interface in stage 7b (Figure 15) by using RCaller. In stage 7c (Figure 15), the Java programming language is used for arrangements and final structuring. In stage 8 (Figure 15), the new SDA and non-spatial data analysis GUIs are integrated into uDig with its extension point. After that, the new menu is added into the uDig menubar. In stage 9 (Figure 15), the designed and created GUI is used for data analysis. If required, the conversion of input and output files is carried out in stages 10 and 2 (Figure 15). In stage 11 (Figure 15), the output of the RScript, which is stored on the local drive, has been read by uDig for display and manipulation. In stage 12a (Figure 15), the result is read by uDig and shown in mapview of uDig for manipulation. In stages 12b and 4 (Figure 15), the result is displayed as an output of the map view of uDig only for reporting purposes.

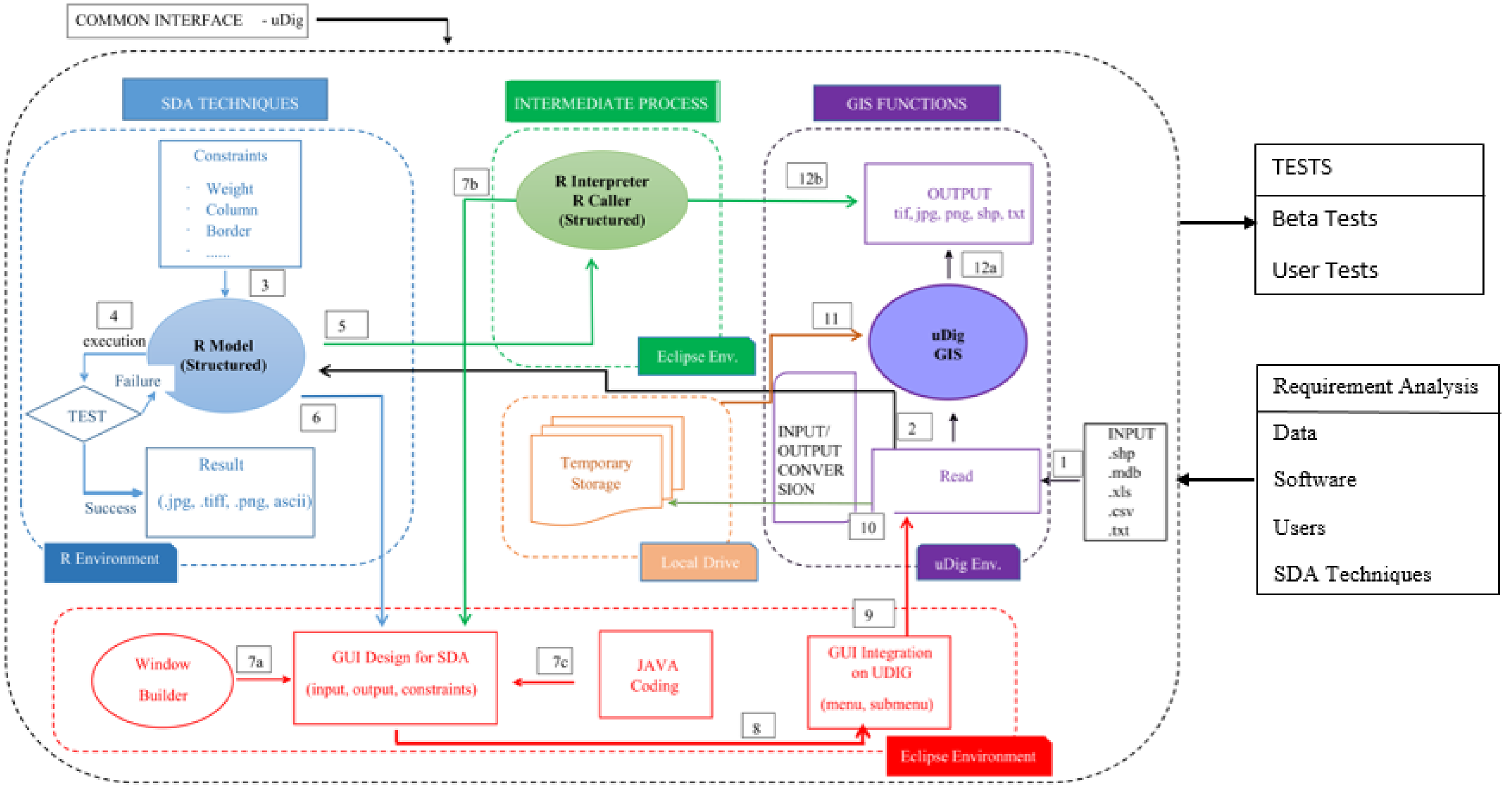


Figure 15. The Graphical Display of the Integration Methodology

4.1.4. Tests

In this study, the users are selected from GIS and SDA users. The user group who has experience with SDA–GIS tools tested the integrated software for three SDA techniques. The testing group involved 32 students from the department of "Geodetic and Geographic Information Technologies" at METU. Then, the integration has been updated with respect to feedback of the users. The user test serves for assessment of the success of the integration. In addition to hands on user test, the setup file and all the related files are shared with the users through a website for an online survey

A web site is designed and created to distribute the source code, documentation, report and survey questionnaires under the domain of www.sda4udig.com. The online testing methodology is helpful in case the community is interested in the integration in uDig and for taking inquiries and feedback. The feedback methodology is passive. In this way, every user who wants to download the source code or the executable software must sign up with their personal contact information.

4.2. The Software Development Life Cycle

Success in developing SDA on GIS primarily depends on the chosen methodology. Therefore, software development needs to be carried out in line with the rules of software engineering. Software product is created when the software requirements are met and these requirements are decided and achieved in terms of software engineering rules and processes. According to the Software Engineering Handbook, software engineering is a type of engineering concerned with the advancement of programming item utilizing decently characterized scientific standards, strategies and methods. The outcome of software engineering is an effective and dependable programming item (Tutorialspoint.com, 2015). Again, it is defined that SDLC is a well-defined, structured sequence of stages in software engineering to develop the intended software product. SDLC provides a series of steps to be followed in order to design and develop a software product efficiently. SDLC framework includes the following steps (Figure 16):

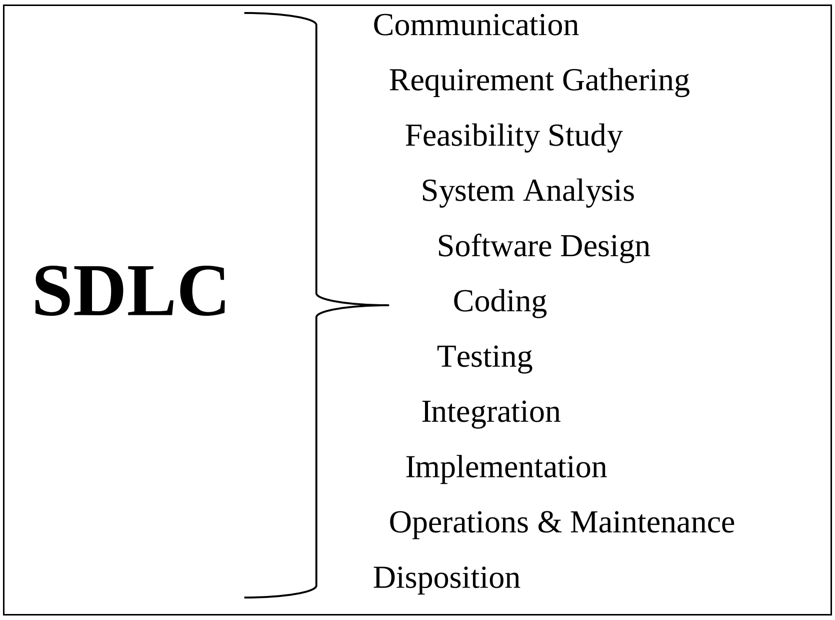


Figure 16. SDLC Framework Steps (adapted from Tutorialspoint.com, 2015)

There are different project lifecycle models in the literature: waterfall, spiral, evolutionary prototyping, incremental development, iterative and incremental development, rapid development, agile development, code and fix, v-shaped, and lightweight methodologies (Wikipedia, 2015). Waterfall is the oldest model in the SDLC engineering process. It is baseline for other models and contains all of the main steps for software development and tests stages. Moreover, it is one of the easiest models to understand and implement. It is also document-driven and the most suitable one for this type of studies (Munassar & Govardhan, 2010).

The adopted waterfall model is shown in Figure 17. Each step of SDLC methodologies is applied in this study and explained in the following sub-sections. Actually, the proposed framework steps depend on using and applying the steps of the waterfall model. Table 7 shows the comparison of the proposed framework and the waterfall model.

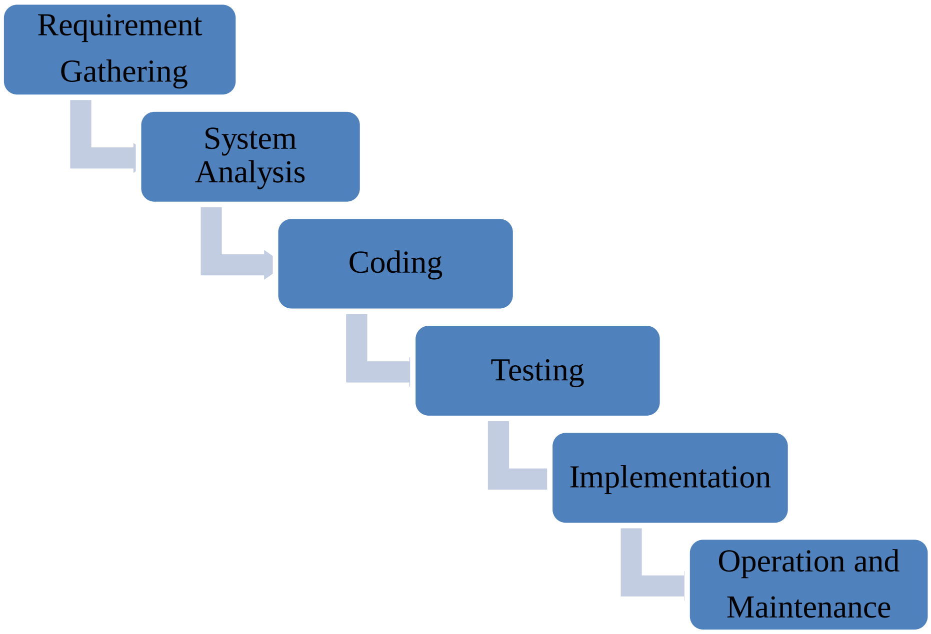


Figure 17. Waterfall Model (adapted from Tutorialspoint.com, 2015)

Table 7. Comparison of Waterfall Model and Proposed Framework

Waterfall Model	Proposed Framework	Explanation of Framework
Requirement Gathering	Requirement Analysis	Both the same process
System Analysis	Determination of Integration Methodology	It is the determination of system parts
Coding	Software Integration	It includes the programming language, environment and coding process. This step includes the implementation of the Waterfall Model as well.
Tests	Beta Test	It is the developer test
Implementation		
Operations&Maintenance	User Test	User test focuses on the end-user feedback and to repair the software integration.

The comparison of this table shows that the framework is suitable for the waterfall model. The following subsections explain each step of the SDLC and how the framework is used for a successful implementation.

4.2.1. Communication, Feasibility Study, Requirement Gathering

The first and the most important step of the software development of life cycle is the requirement analysis. The better the software requirements are, the better software solution is achieved. There are various ways to discover requirements, such as interviews, surveys, questionnaires, task analysis, domain analysis, brainstorming, prototyping, literature review, and observation. Task analysis, brainstorming, observation, literature review and prototyping are the main methods chosen for the requirement analysis in this study. This requirement analysis helps to choose the SDA techniques which are going to be integrated into the uDig. In this step, data, software, hardware, user and SDA techniques requirements are discussed in every detail.

4.2.2. System Analysis

Software analysis and design is the intermediate stage which helps human-readable requirements be transformed into actual code (Tutorialspoint.com, 2015). After the software analysis stage which should conceptualize the software requirements into software implementation. In the determination of the methodology step, the integration type and bridge technique analysis are performed.

4.2.3. Coding

The integration methodology of FOSS SDA with GIS has been decided for the development process.In the software design phase, the integration of SDA techniquesand GIS environment should be decided carefully with respect to the current selected framework. Therefore, the language, development environment, methodology, and implementation strategy should be evaluated and decided in the coding phase of software engineering process. Writing the program code means the initialization of the implementation of a software program, which is to be in a suitable programming language and in an error-free one (Tutorialspoint.com, 2015). Since the uDig is written in Java, an object-oriented language, the coding process is carried out by using the Java language on the Eclipse environment. Since the source code of the uDig is distributed and supported by the uDig community (https://github.com/uDig), it is one of the most important reasons in the selection of the framework in this study. Another reason for the selection of Java is that it is one of the most widely accepted and developed object-oriented language, recently. It allows rapid development as the sources of Java are constantly optimized for years. It is the fastest commercial language and is always up-to-date for convenient usage in the industry (A Cinnober white paper, 2012).

An open-source and widely accepted statistical software of R is chosen as the SDA tool. Modern statistics is performed with statistical computing tools. There is a variety of possibilities such as Minitab, Excel, Stata, SPSS, and R. R is a free software and is available on every platform such as Mac, Windows, and Linux. It is also one of the most reliable and ongoing statistical tools ever developed; therefore, R is chosen as an SDA tool to be integrated to the uDig. As R, the uDig also runs on every platform like Mac, Windows, and Linux. Not only statistical tools, but also R contains SDA and raster image processing packages. It is claimed that R is popular, promotes reproducible research, is up-to-date, contains many packages, is combined with other tools and can be used with other languages such as Java, Phyton etc. (Pruim, 2016). Thus, the R script or languages such as Java should be known in every detail to access a complete integration.

The integration is held by using and knowing Java and R scripts. However, there should be a bridge between R and the uDig, which is RCaller in this study. RCaller is a plug-in that can integrate R with any Java language. RCaller is another way of calling R codes from within Java. It converts data structures to R codes, sends them to an externally created R process, and returns the generated results as Extendable Markup Language (XML), which is the universal way of storing data (Satman, 2014).

4.2.4. Testing

Once the coding phase is completed, the testing phase is critical for getting the right feedback, and according to the feedback from testing, the system design can be revised. In order to discover any errors and to find a solution, it is also critical that a systematic testing should be applied to the developed software.

4.2.5. Implementation

The software needs to be integrated with libraries, databases, packages, and other programs. This stage of the SDLC is involved in the integration of the software with another program. In this study, the R statistical language is integrated with the uDig. There are many different ways to accomplish this integration. Their advantages and disadvantages are discussed in Chapter 2. The RCaller method is chosen to integrate the R and the uDig.

At the integration stage, the question is how these two different components, SDA and GIS, are integrated through an extension or plug-in. An extension is added to the installed software to add new features with an interface. Some other analysis techniques are used through scripting languages such as Visual Basic for applications. Other methods of coupling are also used as a link to GIS with a better processed code for specific usage (Goodchild & Haining, 2003). In this manner, the sda4pp extension for QuantumGIS can be added after installation. On the other hand, the plug-in can be integrated while coding before installation. The GIS industry has recently gained component-based approaches for software by eliminating the reuse of packages, which provided an important advantage in the compatibility of GIS with other software (Ungerer & Goodchild, 2002). The uDig has been suitable for plug-in development, but not for extension so far. Therefore, the plug-in development strategy is required to be adopted primarily for the uDig.

4.2.6. Operation and Maintenance

The remaining part of the software engineering process and the development phase is the maintenance of the software. The feedback from users defines and decides the missing or failed parts of this integration and study. The maintenance focuses on the feedback reported from the end users.

In summary, Table 8 shows the selected parts of the system with their reasons.

Table 8. Component of the System and Summary of their Selection Reasons

System Components	Selected Components	Reasons for Selection
GIS	uDig	Free, Open-source, Compatible with all OS, Java, Sustainable, Compatible with JGrass, Grass, Axios, GeoTools, R and Spatial Tools, GNU/GPL, Portable
SDA Techniques	Structured R Model	Open-source, Sustainable, So many packages, Compatible with uDig, Cross Platform
R Interpreter	RCaller	Compatible with R and uDig, Java, Cross Platform, Sustainable
Development Environment	Eclipse, R Environment	Compatible with uDig, RCaller and R, Includes GUI Designer
Development Language	Java, R Scripts	uDig and R
GUI Developer	Window Builder	Easy-to-develop GUI

All in all, the components of each system have been selected because of those reasons mentioned above.

64

CHAPTER 5

APPLICATION OF THE PROPOSED FRAMEWORK

In this chapter, the framework is implemented to demonstrate its success. The result of the implementation, user surveys and related data analyses as well as considered case studies are given.

5.1. Application of the Proposed Framework

The proposed framework is applied for a real case study examples. The "Requirement Analysis", "Determination of Integration", "Software Integration" and "Tests" steps are explained with regard to implementation.

5.1.1. Requirement Analysis

Requirement Analysis is the first and one of the most crucial steps of the framework in order to sustain the success of other steps and hence the success of the study. As the visualization of GIS is vital for GIS, the input and output and mapview of GIS provide a user-friendly, interactive and easy environment for SDA techniques (Bailey & Gatrell, 1995; Kepoglu, 2011; Duzgun, 2012; Anselin, Syabri & Kho, 2006; Torun and Duzgun, 2008; Anselin, 2000; DeVantier & Feldman, 1993; Massei et al., 2012; Mikelbank, 2010; Anselin & Bao, 1997; GRASS GIS 2008a; Cook et al., 1996; Symanzik et al., 2000). In this way, users can apply SDA techniques with a user-friendly GUI and visualize the input and the output on the mapview of the GIS software. To visualize the input and output on GIS also provide a more interactive environment in order to manipulate both the raw data and final result.

In addition to visualization, the importance and requirements of SDA techniques by GIS is pointed out by various scientists (Goodchild, Haining & Wise, 1992; Goodchild, 1992; Zhang & Griffith, 1997; Anselin, Syabri & Kho, 2006; GRASS GIS, 2008a). This imply that the SDA techniques with the power of GIS increase the understanding of both spatial and attribute data to support the decision makers (Duzgun, 2012; Goodchild, Haining & Wise, 1992; Mikelbank, 2010; Anselin, 2000; GRASS GIS 2008a; Goodchild et al., 2000; Palmer, Bailey & Gatrell, 1996; Star & Estes, 1991; Duckham, Goodchild & Worboys, 2003; Goodchild, 1992).

Since there is a strong requirement to embed the SDA techniques into the GIS software, the developers need a straightforward, easily implemented framework or methodology. For that reason, most scientists, experts and academicians imply the requirement of standardization to implement a framework and methodology (Cremers et al., 1995; Steiniger & Hunter, 2009).

The reason of unsatisfactory SDA and GIS integration is mainly due to the limited number of courses, methodologies and effort (Goodchild, Haining & Wise, 1992; Heywood 1990; Tao, Kainz & Zuidam, 1996; Cook et al., 1996; Symanzik et al., 2000; Anselin & Getis, 1992; Bailey, 1992; Fischer & Nijkamp, 1992; Anselin, Dodson & Hudak, 1993).Therefore, the developers, academicians and researchers have to take initiatives to provide a tool, technique, framework or methodology to decrease the workload of integration for developers and researchers..

The "Requirement Analysis" is composed of five steps, which are "Data", "Hardware", "Software", "User" and "SDA Techniques". Each step is explained in detail in the following sub-sections.

Data:

Even if the ESRI shapefile format is accepted as a standard for the GIS community, the users need to use xls, mdb and csv types of data without conversion. Therefore, a user-friendly GUI is designed and integrated into GIS for the use of the GIS community. Data is not the key factor that affects the success of study, but since it is required for the analysis, it should be accounted for and the expectation of the users should be decided while selecting the other components. Otherwise, incompatibility may cause severe problems during development and usage. Because of these reasons, in addition to ESRI shapefile, xls, mdb, csv and ASCII types of data should be accepted as input by the GIS and SDA software. The selected GIS software is suitable for shp, csv, jpeg and tif types of input and output data. However, the data requirements refer to the usage of xls, mdb and ASCII types of data was well. This type of data usage is compulsory especially for the SDA software. Therefore, xls, and mdb types of input wizard is added to the GIS software by Java coding. Also, ASCII file requirements are met by the integrated SDA software by accepting them as input file format by the SDA software.

To sum up, the integration is able to use formats of xls, csv, mdb, shp and ASCII files as input.

Software:

The selected GIS software is the uDig GIS 1.3.2 and the selected SDA software is R 2.15.3.The GIS interface is currently built on uDig 1.3.2, as shown in Figure 40. uDig runs as a rich client guaranteeing a native look and feel in any of the operating systems. It has a very flexible plugin mechanism to add features and to customize the user interface, but most importantly, it is supported by a very solid industrial foundation (the list of Eclipse members can be found at http://www.eclipse.org/membership/showAllMembers.php). uDig contains not only elements for map visualization but also tools for data manipulation, editing, map printing, and connection with remote databases and servers. Research carried out by the Cascados Project compared the open-source GIS software with respect to the technical perspective shown in Table 9.

Table 9. Technical Potential of FLOS Desktop GIS/RS and GIS Software Products (adapted from Cascados, 2007)

Software	Functionality	Reliability	Usability	Efficiency	Maintainability	Portability	Total
Max. Score	15	9	9	9	9	9	60
QGIS	8.6	3.3	7.4	7.5	7.2	6.7	40.8
uDig	**5.4**	**1.5**	**7.1**	**5.6**	**4.6**	**6.4**	**30.6**
Thuban	6.4	2.4	4.7	6.8	4.0	6.0	30.2
Kosmo	8.1	0.6	4.4	5.4	6.0	4.1	28.6
JUMP	4.2	1.4	3.7	5.4	4.9	6.0	25.6
OpenMap	5.7	3.2	3.5	4.5	4.2	4.2	25.4

In Table 9, the first and the second highest scores in the evaluation criteria are QGIS and uDig for technical potential, respectively. As a result, QGIS and uDig are commonly listed and researched as FOSS desktop GIS software in various literature. There should be some requirements for the GIS software regarding successful implementation criteria such as sustainability, operating system and compatibility. For those purposes, Thuban, QGIS, uDig, OpenJump, Kosmo and Openmap were researched and analyzed with respect to development languages, operating systems, advantages/disadvantages, type of analysis, and suitability and compatibility of integration. The most important selection criterion is the sustainability of the GIS software, which specifies whether the GIS software is continuously developed. uDig has been continuously developed by open-source GIS developers. It is an open-source desktop GIS development platform that includes support for local data, databases, and internet data. The development of uDig started in spring 2004, with initial support from the Canadian GeoInnovations program, and continued throughout 2005 as an independent project (Garnett, 2016).

Additionally, it is essential for this study to select open-source an SDA, GIS and bridge technique. It is easy to download all source code for uDig from https://github.com/udig. Moreover, the compatibility of uDig with JGrass, GRASS, Axios, GeoTools and R are very important advantages for further development.

Its compatibility with multiple operating systems (Microsoft Windows, Macintosh, Linux, etc.) is important for the usage, dissemination and sustainability of the study. In addition, it is compatible with both 32- and 64-bit operating systems.

Four indicators are used to measure the importance of popularity of free GIS software (Steiniger & Bocher, 2009). The fact that a great number of projects have been initiated in the last couple of years is the first indicator. Secondly, financial support by governmental organizations for the foundation of FOS GIS projects increase day by day. High download rates of free desktop GIS software packages are the third indicator. Finally, the increasing numbers of usage cases of open-source GIS software is the fourth indicator (Akbari & Rajabi, 2013).

To sum up, the uDig is selected as the GIS software due to its features listed below:

- Supported by Open Source Geospatial Foundation
- Developed by a broadly-participated international community
- Has GNU/GPL
- Provides development in a plug-in structure
- Easy-to-learn and easy-to-use
- Has open-source development environment
- Has extensive documentation
- Available with detailed and updated manuals
- Supports Java programming language
- User-friendly GUI
- Has cross-platform support
- Has 32- and 64-bit support
- Sustainable and continuously developed

- Has extendible structure
- Compatible with JGrass, Grass, Axios, GeoTools, R and Spatial Tools

The SDA should also be defined and decided carefully to ensure complementary integration. For example, ARC/INFO 7.0.4 already includes kriging and trend surface analysis in its TIN module. ArcView 3.0 is also able to perform network modelling or raster GRID modelling using CAD files (Zhang & Griffith, 1997). Although ArcView includes some simple regression scripts, this is not sufficient for statistical analysis.

As far as the linkage between SDA and GIS is concerned, the current opinion is that even though it seems inappropriate to provide all the possible statistical spatial analysis techniques in GIS, it is feasible and meaningful to integrate the minimum set of SDA techniques,

In this study, the FOSS development strategy is applied for the integration of SDA with GIS. This way is one of the most acceptable methodologies for the development of software and for accelerating dissemination recently. In this way, the development and sustainability of the software strives for a long time without investing any money. Moreover, usage, utilization and testing integrated software are more convenient and realistic. Another reason to use the open-source SDA techniques for integration is the ease of accessing the source code of SDA techniques without a license fee. It is easy to get the source code, to develop and distribute it without any concern. For that purpose, the chosen SDA tools and its techniques should meet some of the requirements for usage, development and dissemination:

- It should work on OS's (Windows, Unix, and Mac)
- SDA languages should work with GIS software compatible
- SDA code should be open-source and there should not be any problems about the distribution of them.
- SDA techniques should be compatible with 32-bit and 64-bit OS's
- SDA techniques are widely used by SDA experts in order to increase the usage and dissemination

- There should be almost all the SDA techniques in order to integrate them in the future if required

R geospatial techniques meet all above mentioned requirements for integration. R version 2.15.3 (Figure 18) version is used for the integration and testing of redesigned R Script in this study.

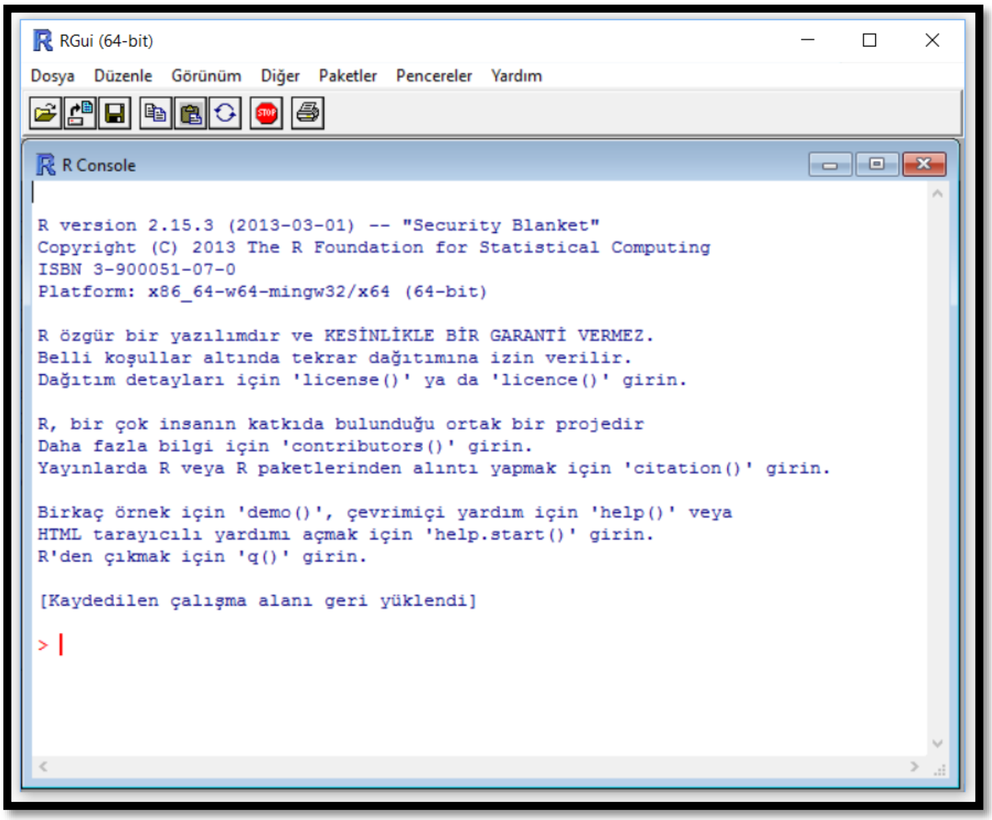


Figure 18. R Environment GUI

There are several concrete studies that successfully integrated R into different types of GIS software for different purposes in the past. Sda4pp is of the studies to integrate several SDA techniques into the GIS environment. It uses the R geospatial techniques as a SDA toolbox integrated into the QGIS software using the Phyton language (Kepoglu, 2011). Another study is the integrating PostGIS, R and QGIS using Phyton and C++ language. For a complementary integration, statistical software is mandatory to perform a geospatial analysis and modelling. R (R Development Core Team, 2009) is probably the best candidate for this task, even compared to commercial systems, given that our modelling might include SR techniques or multi-level modelling (Bonin, 2012).

On the one hand, statistical software is also mandatory to perform tsome data analyses. With respect to those requirements mentioned above, the developed software includes the following properties:

- It performs with the GIS environment properly.
- Various SDA techniques work in the GIS interface properly. The techniques are written in the R language, which is called RScript.
- The development is applied as a plug-in strategy. The plug-in is shared with the uDig Community, which is the development environment for GIS, and also it is shared through the web-site (www.sda4udig.com). The web site is created especially for this study to distribute the source code, plug-in, documents and the exe file for the purpose of usage and dissemination.
- The uDig community and the web-site created are testing environments for this study. In this way, the dissemination of the study is increased.
- GUI is easy to use.
- It supports cross platform, which means that the source code and integration runs on different OS's (Windows, UNIX and Mac).
- There is not any problem about the distribution of both the source code and integration.
- Integration supports multi-languages on GUI.
- There is a user document on GUI as a submenu for easy learning and use.

- The source code is distributed on www.udig.com and www.sda4udig.com for development and modification
- It is free to copy the integration and the source code for any purposes.
- It is easy to install the integration for users.
- The integration includes both the SDA and GIS techniques in the integrated software.
- It is compatible with the GIS data standards.

R is an environment and programming language for data analysis and graphics (Everitt & Hothorn, 2006). R can run on many different OS's such as UNIX, Windows and Mac OS. The base distribution of R and user-contributed packages are licensed under GNU/GPL (Everitt & Hothorn, 2006). R works in accordance with the developed packages by third parties. These packages can be downloaded through different mirror sites in the world. It is added while using the required package, RScripts. The R packages and their documents can be accessed on the Internet (http://cran.r-project.org/web/packages/). Since R is one of the most acceptable SDA tools throughout the world, its SDA techniques are adopted in this study. For those purposes, 14 SDA techniques in R is integrated into the GIS software. Since there is no ready-to-use RScript for these SDA techniques, each technique is searched, developed and tested on the R environment. If each test has successfully run on the R environment, then the integration is initiated on the Eclipse environment. During the integration phase, all the development tests are completed on the Windows OS first, then it is tested on UNIX and Mac OS.

Hardware:

The hardware is not critical in determining the success of the study, but one should define the lowest requirements for the selected software requirements. The selected SDA and GIS software are suitable for most of the currently available computers on the market.

User:

Most of the users, as discussed in the literature section, require a user-friendly GUI to apply the SDA techniques. They want to use more visual software types to obtain deeper understanting of the data. In addition, interactivity is important so that the effects of small changes on the results can be observed. Moreover, they do not want to convert their data to another data type. The study aims to provide a user-friendly GUI for the users; however, the main aims of the study are to provide a framework and methodology for developers. In this way, the developers can integrate any required geospatial technique related to any discipline into the GIS software with a straightforward framework and methodology without excessive effort. The framework and the proposed methodology provide a direction for the developers to design and develop user-friendly geospatial analysis software in GIS software.

To sum up, the users should utilize both the SDA and GIS software to understand the SDA in GIS. In addition, developers must know the logic of the RScripts and also must be at least mid-level Java programmers.

SDA Techniques:

14 SDA techniques of R are integrated into GIS software. Because there is no ready-to-use RScript for these SDA techniques, each technique is researched, developed and tested in the R environment. If each test successfully works on the R environment, then the integration starts on the Eclipse environment. KDE, ADE, UI, Kriging, CSR, FPM, G Function, F Function, K Function, L Function, KST, DE, SR, GWR and 3D Visualization SDA techniques are integrated into the GIS environment. Furthermore, four different non-spatial data analysis R techniques are integrated into GIS. These techniques are histogram graph, pie chart, column graph and summary statistics. R Scripts of all the techniques integrated into GIS is given in Appendix A.

5.1.2. Determination of Integration

To determine the integration, the coupling strategy and bridge technique are two key factors that are evaluated and decided in this step.

Determination of the Coupling Strategy:

As mentioned previously, the highest level of coupling strategy is the fully embedded coupling strategy, but the selected GIS software does not contain any macro language for further development. Therefore, the tight coupling methodology is selected for integration, which is a one-level-lower strategy than the full embedded coupling strategy. Additionally, it fits the requirements of both GIS and SDA techniques. It is required that a common interface to visualize, manipulate and query the result of the analysis is established. Indeed, it is unnecessary to use data or apply analysis in the SDA software. The main objective of this integration is to see, manipulate, retrieve and query the analysis result in the GIS software, which has a common interface. The only deficiency of this strategy is the data conversion, but three different data conversion methods are also embedded into the GIS software to overcome this type of conversion problem.

The integration methodology has the following considerations: (i) to integrate several SDA and non-spatial daa analysis techniques on the GIS software with designing and creating a user-friendly GUI for the users, (ii) most importantly to propose a straightforward methodology to integrate geospatial techniques into the GIS software for at least mid-level GIS developers. To achieve (ii), a unique methodology is proposed with a system architecture in this part. Figure 19 shows the re-design of tight integration methodology with respect to the requirements. It reemphasizes the three aforementioned parts of a tight coupling strategy: GIS, R interpreter and R model. Both R and uDig share the common interface to input, analyze, manipulate and display the results. The structured R model is responsible for SDA and non-spatial data analysis running in the background. RCaller is selected as the R interpreter. The R interpreter is the bridge between the R model and the uDig GIS. Finally, the uDig GIS is responsible for the input, output, display and modification of the data and results.

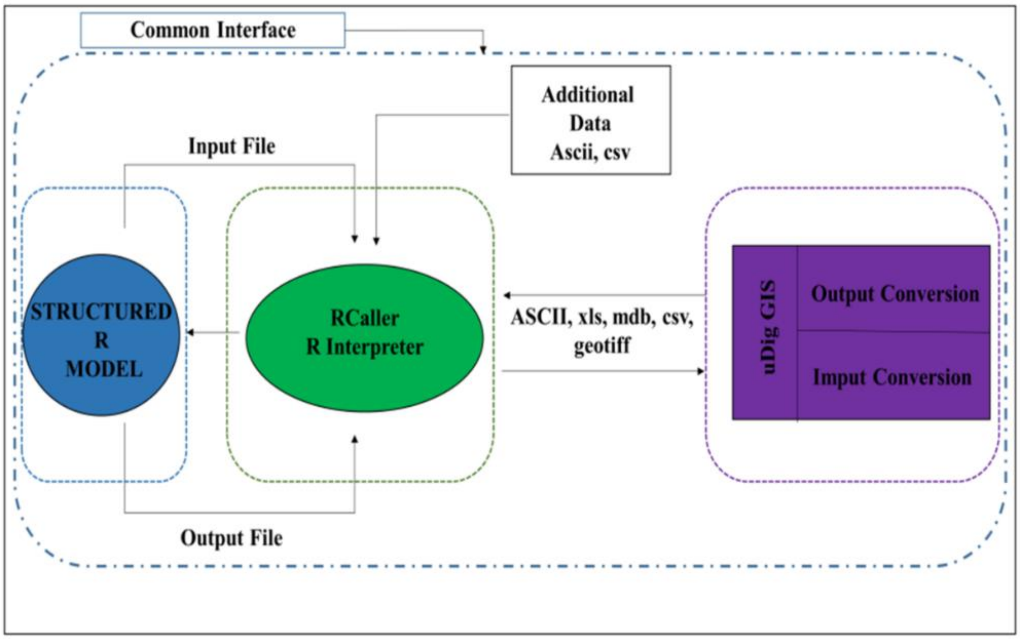


Figure 19. The Graphical Display of the Structured Tight Coupling Strategy

Determination of the Bridge Technique:

RServe is one of the accepted methods for integrating R with other languages. It is an R bundle that permits different applications to converse with R utilizing TCP/IP. It creates an attachment server that can be joined by different applications. RServe provides a flexible structure for developers to integrate R with several other programming languages such as C/C++, Java and PHP. It is able to connect interfaces between applications and R, which is faster, more language independent, and more remote capable than any other program. R and the application (server) are separated well, so there are no internal data manipulations affecting each other (Urbanek, 2003). RServe can be introduced in both Windows and Linux platforms, although it is prescribed for introduction and utilization on a Linux platform (RServe, 2015). One of the biggest disadvantages of RServe is that it can be installed only on Linux. The requirement of a TCP/IP connection is another prominent disadvantage. When the connection is lost, the software may encounter severe problems.

Another solution is a console-based method called **VASMM**. A console-based interface, in which commands for R are stored in a file and written into another file, is currently used by several applications, such as VASMM (Tarumi, Iizuka, Mori & Tanaka, 2003). Although VASMM is one of the solutions to connect R with other languages, it has certain problems in the connection of R with Java, making it useless for integration of R with uDig. VASMM is also relatively slow because each instance of R must be initiated with each request.

SJava is another solution to connect R with Java and vice versa. It allows one to call R from Java and Java from R. It works only locally because it embeds R into Java via JNI. Because there is no synchronization between Java and R, and given that R is not multithread safe, it is fatal to make more than one call from Java to R.

Another integration technique of R and Java is **JRI**, which allows R to run inside Java applications as a single thread. Basically, it loads the dynamic library of R into Java and provides a Java API for R functionality. JRI is the inverse of rJava. It works on all operating systems, but it supports simple calls to R functions. Therefore, for some complex analyses, R scripts cannot be called with JRI.

Another method is using **Renjin** for the integration of R and JVM. This integration supports R in terms of both application and database. This method is defined on the Renjin website as a new implementation of the R language and environment for the Java Virtual Machine (JVM), which can transparently analyze large datasets and integrate them with other systems, such as databases and application servers. As an advantage, Renjin will be implementable in C and FORTRAN languages; however, this is currently being developed (Renjin, 2015).

Finally, **RCaller** is a programming language and environment that is mainly aimed to be used for statistical calculations and data analysis. RCaller translates data into R codes, transmits them to an R process, and then recalls them in XML format, which is a universal method for saving data (Satman, 2014). There is no setup process for RCaller; it is only a Jar file (Renjin, 2015). RCaller works on all operating systems. It can provide both simple and complex RScript syntaxes and Java interactions. The performance of RCaller includes the creation of an external Rscrpit.exe process, so the interaction time is high as expected. Moreover, it allows plotting of graphs. Satman (2014) applied a performance test with some of the rJava tools. In the test case, an integer array x is created with a size of 1000. This array is then passed to R, where x2 values are calculated and then the new array is handled by Java. While the operations performed in the R side do not affect the performances of libraries, no larger calculations are preferred. Operations are iterated 100 times for each library. Performances of algorithms are shown in Table 10.

Table 10. Descriptive Statistics of Times Consumed by Algorithms (in milliseconds) (Satman, 2014)

	RCaller	RCaller Online	RServe	rJava
Min	557.00	257.00	0.00	0.00
Max	643.00	296.00	14.00	9.00
Mean	596.90	266.96	1.21	1.28
Median	565.00	263.00	1.00	1.00
Std.Dev.	14.92	9.63	1.76	1.39
MAD	4.45	5.94	0.00	1.48

In Table 10, it is shown that the rJava and RServe outperform the RCaller by means of interaction times. The performance of RCaller includes the creation of an external Rscrpit.exe process, so the interaction time is high as expected.

Hence, RCaller is used as a bridge between R techniques and uDig because of its suitability, performance and the other criteria mentioned above. At the integration part, the question is whether these two different parts, SDA and GIS, should be integrated through an extension or a plugin. uDig is suitable for plugin development but not for an extension. Therefore, a plugin development strategy must be applied to uDig primarily.

5.1.3. Software Integration

The software integration step is composed of "Definition of the Architecture of the System", "Selection of Programming Language(s)", "Selection of Development Environment" and "Implementation of the Methodology". Each step is explained in great detail in the following sections.

Definition of the System Architecture:

After deciding the requirements of SDA, GIS and Bridge Technique, the system structure should be figured out and explained.

Figure 20 shows the architecture of the system. uDig is developed on the concept of plugins to the base Eclipse Rich Client Platform. In addition to using and extending its tools and techniques, it uses several support libraries such as GeoTools, Axios, JGrass, Grass and various extensions to the Java Runtime Environment. To extend the function of uDig, there should be some new input formats such as XLS and MDB that can be integrated via the Java programming language. SDA and non-spatial data analysis plugins can also be developed on the Eclipse environment to support uDig, with respect to spatial and non-spatial data analysis, by integrating R using RCaller. R contains various packages for analysis with respect to different requirements of various disciplines. Each package is added and used by R while implementing them into uDig. RCaller works as a type of bridge between R and uDig to integrate all geospatial and non-geospatial techniques into uDig. To diminish the coding workload for developers, Window Builder can be used for interface design and development.

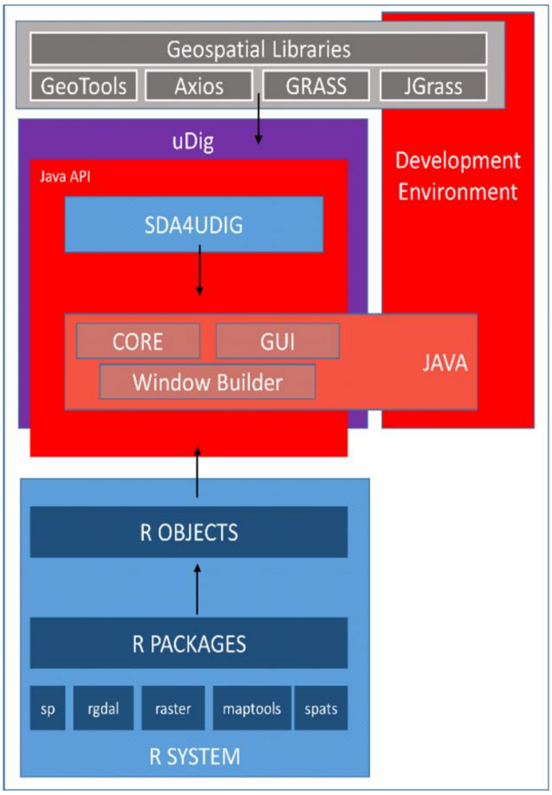


Figure 20. Architecture of the System

The architecture of sda4uDig consists of an interactive environment that combines maps with graphs and SDA results. The current version uses the ESRI shapefile as the standard for storing spatial information. It contains functionality to read and write such files, as well as to convert ASCII text input files, Ms Word Access files and Ms Word Excel files for point coordinates or boundary file coordinates to the shapefile format. The analytical functionality is implemented in a modular fashion, as a collection of modular packages with associated methods.

In broad terms, the functionality can be classified into six categories:

- spatial data manipulation and utilities: data input, output, and conversion,
- data transformation: variable transformations and creation of new variables,
- mapping: classified maps,
- Exploratory Data Analysis (EDA): statistical graphics,

- spatial autocorrelation: global and local spatial autocorrelation statistics, with inference and visualization,
- spatial regression (SR): geographical weighted regression, linear SR models.

The full set of functions is listed in Table 11 and is documented in detail in the sda4uDig user's guides. The software implementation consists of two important components: the user interface and graphics on the one hand and the computational engine on the other hand. The computational engine (including statistical operations, geospatial analysis, and spatial regression) is RScript code and largely cross platform.

Table 11. The sda4uDig Functionality Overview

Category	**Functions**
Spatial Data	Data input from shapefile (point, line, polygon) Data input from text file (point, line, polygon) Data input from csv file (point, line, polygon) Data input from mdb file (point, line, polygon) Data input from xls file (point, line, polygon)
Data Transformation	Variable transformation Queries
Mapping	Quantile, Natural Break, Equal Interval, Unique Value Maps Filtered Maps Classified Map both for vector data and raster data 3D visualization of raster maps
EDA	Histogram Pie Chart Column Graph
Spatial Analysis	Kernel Density Estimation Adaptive Density Estimation Moran's I Geary's C K, L, G, F Functions CSR Buffered Density Estimation Kriging Uniform Intensity Komolov Smirnov Test Fitted Poisson Model
Spatial Regression	Spatial Regression Model Geographical Weighted Regression Model

Selection of the Programming Language(s):

The R software uses RScripts, hence developers should be familiar to the RScripts. Because uDig is coded with an object-oriented programming language (Java), the integration is implemented with the same language and thus the developers need to be familiar to Java as well. As RCaller is written in the Java programming language, the integration is easier and compatible. Table 12 list the ranking of various programming languaes and the position of Java programming languages is remarkable. Since Java is one of the most acceptable languages in the world withmillions of software developers, the selection of Java languages as the development languages enhance this study's sustainability.

Table 12. Position of Top 10 Programming Languages for 5, 15 and 25 Years Ago (TIOBE, 2015)

Programming Language	**Position**			
	Aug 2010	**Aug 2005**	**Aug 1995**	**Aug 1985**
Java	**1**	**1**	**-**	**-**
C	2	2	3	1
C++	3	3	2	11
PHP	4	5	-	-
(Visual) Basic	5	6	1	4
C#	6	7	-	-
Python	7	8	24	-
Perl	8	4	8	-
Objective-C	9	43	-	-
Delphi	10	10	-	-
Lisp/Schema/Closure	16	14	6	2
Ada	29	17	7	3

Selection of the Development Environment:

As uDig is developed in the Eclipse Indigo environment, and RCaller is also suitable for RCaller development and RScript integration, the Eclipse Indigo version is selected for the development of integration.

The integrated software should be tested continuously on an OS in each step. Therefore, the Windows 64-bit OS is selected as a test platform. After successful implementation on Windows, then other platforms are also tested. Since the communication of R techniques and the uDig is achieved by using the RCaller, it should be restructured with respect to the SDA and GIS requirements.. Finally, the R techniques should also be debugged and run on a development environment before implementation. The R environment is suitable for this purpose. Furthermore, the Eclipse Indigo development environment is also suitable for testing R techniques using the RCaller communication channel.

The Implementation of the Proposed Methodology:

In this part, a case study is implemented to follow the steps of the methodology. The case study is the “Column Graph”, which is a non-spatial dta analysis techniques. The critical part of the source code in order to define and explain the implementation methodology is given below.The entire source code is shared through the website.

<u>Step 1:</u>

The data is added into the uDig GIS software.

<u>Code, GUI and Process of Step 1:</u>

Figure 21 shows the shapefile data on the mapview of the uDig.

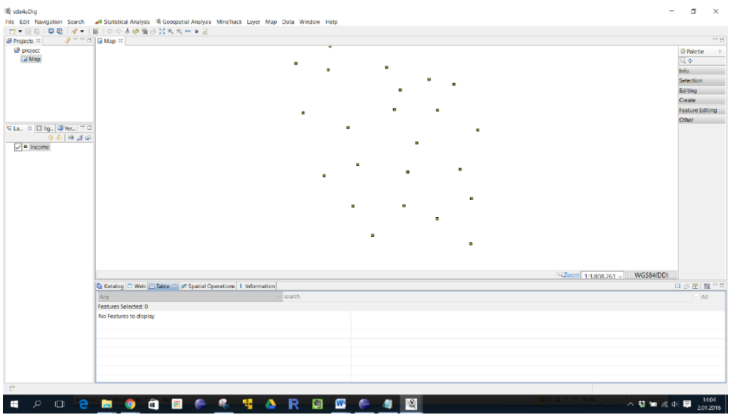


Figure 21. Data in MapView of uDig

Step 2:

If the data is csv, mdb or xls type, the import wizard is implemented to convert data into the shp file type.

Code, GUI and Process of Step 2:

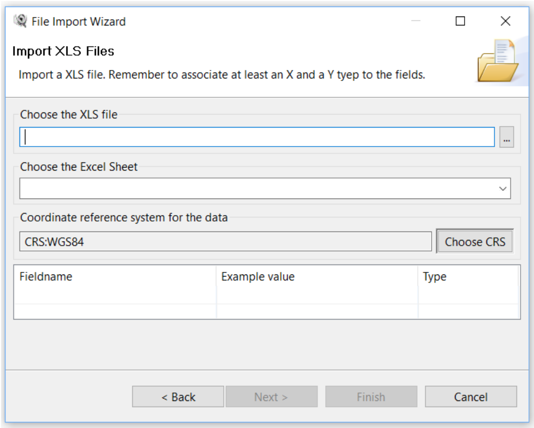


Figure 22. Import xls File into Shapefile Wizard

Step 3:

Since there is no ready-to use R Script, the definition of each technique including input, output, constraints and other variables has been achieved in this step.

Code, GUI and Process of Step 3:

The constraints are defined considering the requirements of SDA technique and usersfor this purpose, one can use the documentation of each package distributed by the third parties

Result of Step 3:

The sample technique requires the column, breaks number, col, labels, x and y axis, title and border definitions. The definition of the technique with its input, output and parameters are shown in Table 13.

Table 13. R Script of Column Graph

install.packages(“tensor”) install.packages(“abind”) install.packages(“deldir”) install.packages(“nlme”) install.packages(“mgcv”) install.packages(“lattice”) install.packages(“raster”) install.packages(“spatstat”) install.packages(“sp”) install.packages(“maptools”) install.packages(“foreign”) install.packages(“rgdal”) require(tensor) require(abind) require(deldir)	require(nlme) require(mgcv) require(lattice) require(raster) require(spatstat) require(sp) require(maptools) require(foreign) require(rgdal) dataOgr <- readOGR("C:/udigData/data" ,"Income") analysis <- hist(dataOgr$incomeTLpe, breaks=8, col=2, labels=TRUE, xlab="X Axis of Graph", ylab="Y Axis of Graph", main="Column Graph of Data", border=3)

Step 4:

Each geospatial technique is tested on the R environment. In this way, the input, output, parameters and other variables can be defined. In addition, the result and success of the study is proven in this step. If the result is successful, the next step is applied; otherwise, the R Script is iterated until successful results are obtained.

Code, GUI and Process of Step 4:

The process includes the R package download and setup. Then the structured R Script coming from step 3 is tested on R environment.

Result of Step 4:

Figure 23 shows that R Script for this technique is successfully working.

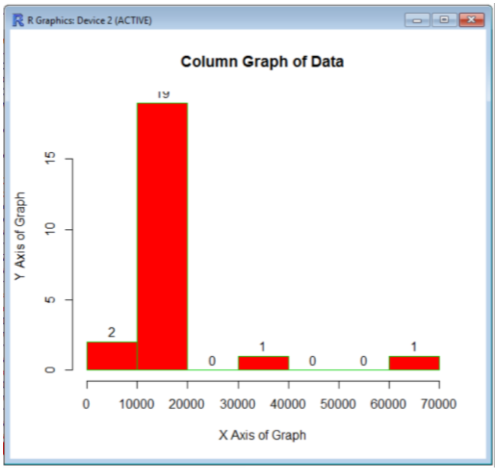


Figure 23. Result of Column Graph

Step 5:

The structured R Script which is designed and defined in step 4 is integrated with RCaller and tested on Eclipse environment before embedded into its GUI.

Code, GUI and Process of Step 5:

RCaller is used for testing RScript on eclipse environment whether it is returns the same result within step 4 or not.

Result of Step 5:

The result is the same with in Figure 23.

Step 6:

The definition of the input, output, parameters and other variables related with the technique are finalized at this step.

Code, GUI and Process of Step 6:

The definition of the parameters are definedwith respect to the requirements of the SDA technique and user. The Table 14 shows the GUI design requirements

Table 14. GUI Design Parameters for Column Graph Example

Parameters	Definition
Input	Shape Data
Column	Income
Breaks Number	8
Col	Red, green
Label	True, false
X and Y Axis	X and Y axis of the graph selected and defined
Title of the Graph	Column Graph of Data
Output	Jpeg, png

Result of Step 6:

The example technique requires the column, breaks number, col, labels, x and y axis, title and border definitions.

Step 7a:

The Window Builder is used to design the GUI for each technique.

Code, GUI and Process of Step 7a:

Figure 24 shows the Window Builder interface integrated into the Eclipse Indigo.

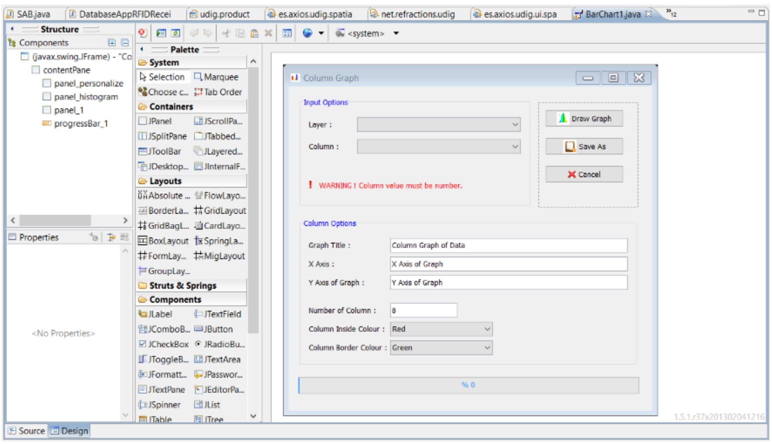


Figure 24. Window Builder Interface for Column Graph Design

Result of Step 7a:

The Window Builder Plug-in is especially suitable for mid-level developers. Moreover, it decreases the workload of developers to design and create a GUI for each technique. Figure 25 shows the design of column graph created by the Window Builder.

Step 7b:

If the data is shapefile, it is sent to R via the RCaller. R technique(s), which is integrated R Caller with a GUI, are tested in this step.

Code, GUI and Process of Step 7b:

The designed GUI is tested in this step with the integration of the RCaller. The code part is shown in Table 15. The RCaller and RCode are two main methods in the RCaller interpreter to send the RScript to the R environment to be executed.

Table 15. A Part of Code for Execution of R Script with the Designed GUI

```
{
RCaller = new RCaller();
            RCode = new RCode();
            RPackages.loadRPackageHistogram(rCaller, rCode);
            RPackages.setRPackageHistogram(rCaller, rCode);
            plotFile = rCode.startPlot();
            lineDataOgr = "dataOgr <- readOGR(\"" + SAB.outputSaveDirectoryURI + "\" ,\"" +
selectedLayer + "\")";
            lineAnalysis = "analysis <- hist(dataOgr$" + selectedColumn + ", breaks=" +
selectedNumOfColumns + ", col=" + selectedFillColor + ", labels=TRUE" + ", xlab=\"" +
selectedXTitle + "\""
                  + ", ylab=\"" + selectedYTitle + "\"" + ", main=\"" + selectedGraphTitle + "\"" + ",
border=" + selectedBorderColor + ")";
            rCode.addRCode(lineDataOgr);
            rCode.addRCode(lineAnalysis);
            rCaller.setRCode(rCode);
            rCaller.runOnly();
            rCode.showPlot(plotFile);
         return null;
      }
```

Result of Step 7b:

The RCode sets up the RScripts; the RCaller sends the RScripts to the R executable directory to be executed. The RPackages contains all the packages required for all the techniques. This class contains two main methods, which are loadRPackage and setRPackage. The load RPackage installs the required R Packages. The setRPackage method also sets the RPackages, which are essential for each technique during execution.

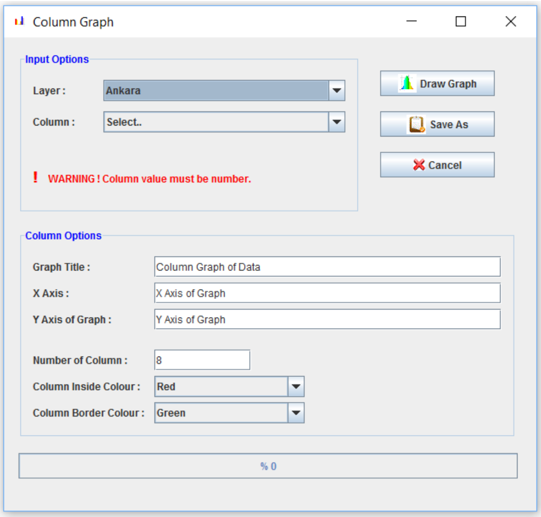


Figure 25. Column Graph Design with Window Builder

Step 7c:

In this step, the Java programming language is used to finalize the design of the GUI.

Code, GUI and Process of Step 7c:

Table 16 shows some Java codes to finalize the GUI design of each technique.

Table 16. A Part of Java Codes GUI Finalization, Handler Design

```
public static void main(String[] args) {
      EventQueue.invokeLater(new Runnable() {
         public void run() {
            try {
               BarChart hist = new BarChart();
               hist.setAlwaysOnTop(true);
            } catch (Exception e) {
               e.printStackTrace();
            }
         }
      });
   }
```

<u>Result of Step 7c:</u>

Step 7c is only to design and create a GUI for each technique but each technique requires some more Java codes to implement and finalize the technique.

<u>Step 8:</u>

In this step, the extension point of the uDig is used to integrate the R technique into the uDig.

<u>Code, GUI and Process of Step 8:</u>

Table 16 shows some Java codes to finalize the GUI design of each technique.

Table 17. A Part of Java Codes for GUI Finalization, Calling Column GUI

```
public class barChartHandler implements IHandler {
   public Object execute(ExecutionEvent event) throws ExecutionException {
      IWorkbenchWindow window = HandlerUtil.getActiveWorkbenchWindowChecked(event);
      try{
         HistogramChart hc = new HistogramChart();
         hc.setVisible(true);
      }catch(Exception e ){
         e.printStackTrace();
      }
      return null;
   }
```

Figure 26 shows the extension point of uDig menu and submenu.

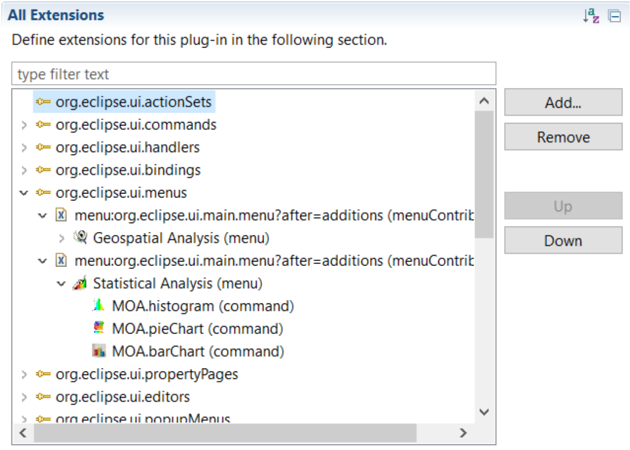


Figure 26. Adding Menu and Submenu on the uDig

Result of Step 8:

The development structure of the uDig allows developers to use the Extension Point to integrate any developed geospatial and non-geospatial technique into the uDig.

Figure 27 shows the menu and submenu on the uDig menu bar.

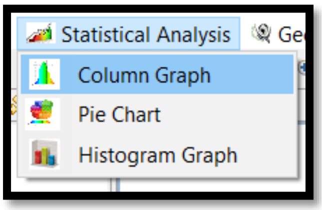


Figure 27. Menu and Submenu on the uDig

Step 9:

In this step, the final GUI of each technique can be used for entering input data for analysis.

Code, GUI and Process of Step 9:

The input, output and other remaining parameters related with each technique are selected on GUI of each technique for analysis purposes.

Result of Step 9:

Figure 28 shows the GUI of column graph and parameter definition of this technique.

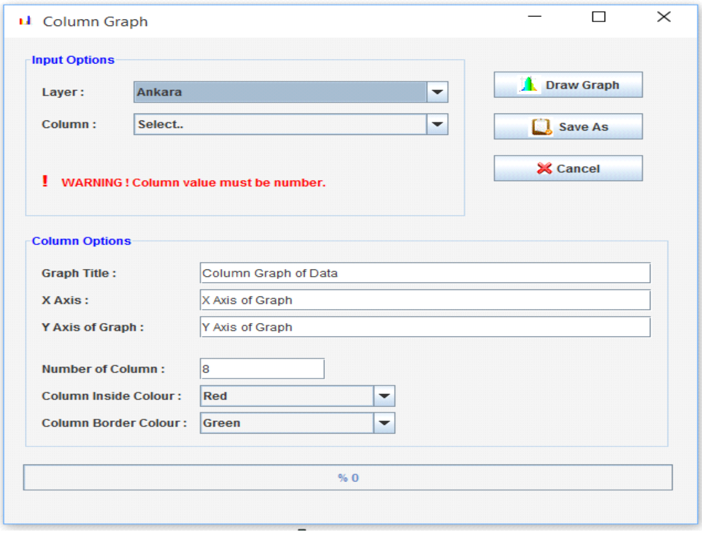


Figure 28. The GUI of Column Graph and the Technique Parameter Definitions

Step 10:

The result of the technique is sent to the created directory and stored there. The result can be converted into the correct type which has been accepted by the uDig as input and output.

Code, GUI and Process of Step 10:

Table 18 shows the result of geospatial and non-geospatial technique stored in a directory.

Table 18. A Part of Code for the Storage of the Result of Technique

```
String selectedLayer = comboBox_PointLayer.getSelectedItem().toString();
        String selectedFraction = spinner_Fraction.getValue().toString();
        String outputTitle = selectedLayer + SAB.createDateTag();
        String outputName = "";
        if (comboBox_RasterFormat.getSelectedItem().toString() == "GeoTIFF (.tiff)") {
            outputName = outputTitle + ".tiff";
        } else if (comboBox_RasterFormat.getSelectedItem().toString() == "JPEG (.jpeg)") {
            outputName = outputTitle + ".jpeg";
        }
        String outputDirectory = SAB.outputSaveDirectoryURL + outputName;
        URL saveOutputDirectory = new URL(outputDirectory);
        final File outputFile = new File(saveOutputDirectory.toURI());
```

Result of Step 10:

If the result should be shown on mapview of the uDig, it is stored in this directory; otherwise, it does not need to be stored there.

Step 11:

This step is the visualization of the analysis result on the mapview of the uDig.

Code, GUI and Process of Step 11:

The geospatial analysis is shown on the mapview of the uDig. Although the geospatial result can be shown in the mapview, it can also be shown as screenshot out of the uDig mapview. The code is shown in Table 19.

Table 19. A Part of Code to Call the Result of Geospatial Technique on the Mapview of the uDig

```
String lineWGdal = "";
        if (comboBox_RasterFormat.getSelectedItem().toString() == "GeoTIFF (.tiff)") {
            lineWGdal = "writeGDAL(as.SpatialGridDataFrame.im(analysis),\"" + outputFile.toString().replace('\\', '/') + "\",\"GTiff\")";
        } else if (comboBox_RasterFormat.getSelectedItem().toString() == "JPEG (.jpeg)") {
            lineWGdal = "plot(analysis)";
        }
if (comboBox_RasterFormat.getSelectedItem().toString() == "GeoTIFF (.tiff)") {
            rCaller.setRCode(rCode);
            rCaller.runOnly();
            IMap map = ApplicationGIS.getActiveMap();
            LayerFactory = map.getLayerFactory();
            IProgressMonitor monitorr = ProgressManager.instance().get();
            IServiceFactory factory = CatalogPlugin.getDefault().getServiceFactory();
            java.util.List<IService> services = factory.createService(saveOutputDirectory);
            services.get(0).resources(monitorr);
            analysisOutputLayer = layerFactory.createLayer(services.get(0).resources(monitorr).get(0));
            AddLayerCommand cmd = new AddLayerCommand(analysisOutputLayer);
            map.sendCommandASync(cmd);
        }
```

Result of Step 11:

Since the example technique is not a geospatial one, the analysis result is not shown in the mapview of the uDig.

Step 12a:

The result which is created with the GUI that has been designed and created can also be read by the uDig in this step.

Code, GUI and Process of Step 12a:

There is no code in this step. The output of each technique is visualized on the mapview of uDig for further manipulation and designing at this step.

Result of Step 12a:

The result of this step is the output of the analysis.

Step 12b:

This step is the visualization of the analysis result on out of the mapview of the uDig.

Code, GUI and Process of Step 12b:

The non-geospatial analysis is shown on out of the mapview of the uDig. The code line showing the analysis out of mapview is shown in Table 20 .

Table 20. A Part of Code which Shows Analysis Result out of the Mapview

```
rCode.addRCode(lineDataOgr);
rCode.addRCode(lineAnalysis);
rCode.endPlot();
rCaller.setRCode(rCode);
rCaller.runOnly();
            rCode.showPlot(plotFile);
```

Result of Step 12b:

The result of the column graph is shown in Figure 29. This result shows the success of the proposed methodology.

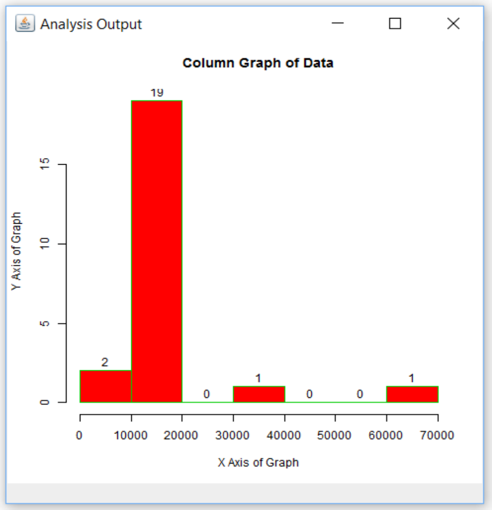


Figure 29. Result of Column Graph Shown out of Mapview

5.1.4. Tests

The test procedure is applied using the SDLC which is explained and discussed in Chapter 4. Mainly two types of tests are applied which are the alpha and the beta tests. The alpha test is applied by the developer and the beta tests are applied to the user of GIS and SDA in two different ways. To get feedback form users and to increase the dissemination, a survey is prepared and distributed through the website for the visitors. The integrated software and survey are provided for on-line users. They were asked to fill the questionnaire after using the software. The second one is questionnaire survey performed to the users who conducted hands-on. Exercise using the developed software.

Alpha Tests:

The development of sda4uDig integration should be debugged and run in every step of coding. Therefore, the step of testing in the SDLC is applied for each development. During the development, the coding step in the study is as follow:

- Development of the R techniques
- Integration of the R techniques with the RCaller
- Designing GUI for each technique
- Integration of the R-RCaller integration with its GUI
- Integration of the R-RCaller-GUI with the uDig

Each step mentioned above should be tested by adding even a single line of code after development. Therefore, in addition to the SDLC testing procedure, each step mentioned above is tested one by one after the development process.

Development of the R techniques is actually the first step of the algorithm development and the coding phase. Currently, there is no ready-to-use R technique in detail but generally it is possible to find the R technique. Therefore, each technique is developed with respect to different requirements. Because of this reason, each technique is developed and tested iteratively on the R environment first. After successful results, the R technique and its RScript is integrated with the RCaller.

Integration of R technique with RCaller is achieved on the Eclipse development environment. The RCaller can connect Java with R but it needs some designing and update. In this way, the R technique is sent to the R environment running on the background without user interface.

Once the integration of R and the RCaller is performed successfully, the ***GUI design*** and coding phase is now initiated. The design of each user interface for each technique is developed with respect to input, output and other constraints. The GUI is also tested even if the R-RCaller integration works successfully or not.

The next phase after the R-RCaller integration and the GUI design is ***the integration of R-RCaller and GUI*** of each technique. This test step shows that the RScript technique works successfully with its GUI. This test is applied a couple of times with different data. The data is changed with different size and type in order to see the malfunctions.

Finally, the successful ***integration of R, RCaller and GUI is now being integrated into the GIS software*** with the uDig extension point. Once the integration is completed successfully, the technique on the uDig interface is tested with different data in regard to size and type. Even though there are no malfunctions, the test is iterated several times. In addition, though one single code is added to the R Script or any phase of integration, the testing phase is iterated from the first step to final step.

Finally, the successful ***integration of R, RCaller and GUI is now being integrated into the GIS software*** with the uDig extension point. Once the integration is completed successfully, the technique on the uDig interface is tested with different data in regard to size and type. Even though there are no malfunctions, the test is iterated several times. In addition, though one single code is added to the R Script or any phase of integration, the testing phase is iterated from the first step to final step.

Beta Tests:

In this part, questionnaires are developed and a survey is conducted in two different ways: online and hands-on manner. The test methodology is developed first to understand the success of the study and then to understand the users' approach and satisfaction.

First of all, the ***hypotheses*** are constructed to test the success of the study. Once the hypotheses are constructed, the ***questionnaire*** is prepared to collect data to test the hypotheses that have been constructed. Then, a ***user task*** is prepared for the users to test the integrated study.

Finally, two ***survey questionnaires are applied*** for different user types. The first user type is online users who have reached this study through the web site. The second one consists of those users who are students at the GGIT department at METU. The survey questions are prepared with respect to the hypothesis that is analyzed in order to give some useful statistics to prove the success of the study and to show a direction for developers and researchers to do further research or update malfunctions of the integrated software.

- Hypothesis

Based on the aim and objectives, the following hypotheses are determined in Table 21. Each hypothesis is analyzed with a different test. These hypotheses are to measure the success of the study and to get feedback and insight of users.

Table 21. Hypotheses about "Integration of Geospatial Techniques (R) into an Open-Source GIS Software (uDig)"

NO	HYPOTHESES	Test Method
H1	There is a relationship between usage frequencies of the integration with the success level of GIS-SDA diffusion.	LR*
H2	There is a relationship between the satisfaction and the quality of GIS-SDA integration.	ANOVA
H3	There is a relationship between satisfaction and the documentation of the GIS-SDA integration.	ANOVA
H4	There is a relationship between satisfaction and the usability of the GIS-SDA integration.	ANOVA
H5	There is a relationship between satisfaction and the performance of the GIS-SDA integration.	ANOVA
H6	There is a relationship between satisfaction and the easiness of the GIS-SDA integration.	ANOVA
H7	There is a difference between users with different number of geospatial techniques in terms of the satisfaction of the GIS-SDA integration.	ANOVA
H8	Visualization, SDA techniques, performance, GUI, effectiveness, utility and time are main reasons for users to prefer and use the GIS software.	Descriptive

*LR: Linear Regression

- Questionnaire

After defining the hypotheses to test the success of the study, the questionnaires are prepared for different user types. The website is designed and developed in order to share all the documents and source codes for dissemination. This is not only for dissemination but also to get feedback from users through the questionnaires.

The first questionnaire is prepared for visitors to get only simple statistics about their professions and contact information. This questionnaire is given in Appendix B. The questions are shown in Figure B 1.

The second questionnaire is prepared to collect more feedback from the users of integration. Therefore, this questionnaire is filled by software users after using it. The study is updated with respect to the insight of users. The the answers to the questionnaire applied for hands-on usersare analyzed as they provide answers after using the software. This questionnaire is attached in Appendix C. The questions are shown in Figure C 1.

- Users Task

Once the hypotheses are constructed and the questionnaires are prepared, a user task is prepared with three different case studies. It is assured that the three cases should be different with their application. Therefore, Kernel Density Estimation (KDE) and K Function are selected as SDA techniques. The third one is selected as a non-spatial data analysis technique, which is a column graph. Users follow each step of the tasks one by one and apply them to get the same result within the prepared task. The task is attached in Appendix D. This appendix contains figures from Figure D 1 to Figure D 14.

Surveys:

Online Survey:

To increase the usage, distribution and dissemination of the study, web application is the most suitable environment to reach the SDA and GIS community who is interested in such software and study types. The users of the SDA system are examined in order to understand how much interest is seen from the users and who uses the system. For that purpose, a website is developed for this study to distribute the user documents, plug-in, source code, user application usage videos, articles and related results and documents. The official website for the study is http://www.sda4udig.com.

The website of the study helps provide a complementary solution for GIS and spatial data users. For that purpose, the feedback of users whether they are GIS experts or spatial data analysts is important to understand if the integration successfully works or not. This way is also very helpful for the developer to catch the malfunction of the software, which could be only understood with the developer test. In addition, it is critical and important to collect the idea of users for further development of the study. In order to do that, there may be various methodologies to get feedback of the users but the application of questionnaire, online test and hands-on tests are applied in this study. These testing methodologies ensure the sustainability of the study as well, because it increases the distribution of the study through the developers and users.

The participants of the online test are certainly website visitors and users. Two types of audiences are targeted again to test the sda4uDig integration. It is the main aim of these surveys to get feedback from users; therefore, there are two questionnaires in this website under the “Form” sections, which are “Survey” and “Survey for Users”. The “Survey” is prepared to collect the user and developer data to show certain statistics. On the other hand, “Survey for User” is used to collect the feedback from the users and developers to keep the study sustainable and living.

To collect the user information, the users have to fill at least the Survey questionnaire to download the setup file. However, the Survey for Users is optional for users. Statistics about the registered users have been collected through this website. The user statistics have been collected by Google Analytics. With respect to these statistics, 436 users registered between July 2015 and February 2016. It is assumed that these 436 users have downloaded and used these integrated software. According to the statistics between July 2015 and February 2016, the number of absolute unique visitors who visited the web site is nearly 588. These figures are explained in detailed in the following sections.

Hands-on Survey:

32 users are included in this survey and all of them have responded to the questionnaire. These users have been chosen from the GGIT department in two different lectures: one is the “Principles of Remote Sensing” and the other is the “Introduction to GIS”. 14 users have set up and used the sda4udig with a

documentation provided for them in the Principles of Remote Sensing course. Furthermore, eight users have set up and used the sda4uDig with a documentation provided for them in the Introduction to GIS course. In addition to these 22 users, the integration has been tested by 10 more GIS and SDA users and they have completed the questionnaire after usage.

The data collected from this survey is analyzed by using descriptive statistics, reliability tests, linear regression analyses, and ANOVA to test the hypotheses and provide findings. In addition, some graphical representations are used to explain the use of sda4uDig software users. To test the users' questionnaire survey results, the SPSS 15 statistical analysis tool has been used.

- Hypothesis Testing Based on Questionnaires

The statistics about the users of sda4uDig who visit the website are recorded by Google Analytics. Several free packages are embedded in the home page of the website for statistics and questionnaire. These packages create a mechanism for the Google Analytics service to collect the information about the user who has entered and registered at the website. Statistics have been recorded between July 2015 and February 2016. Google Analytics has provided the statistical figures about the number of absolute unique visitors, visits, average visits per day, countries where the visits come from, visit page number, average time spent in the site by the visitors, and returning and new visitors.

Online User Test Analysis:

The total number of absolute unique visitors to the website is 588. The distribution of the number of visits by countries is shown in

Figure 30. The highest number of visits was from Turkey with 134 times. Top six countries can be listed as Turkey with 134 visits, Japan with 60, United States of

America with 43, Germany with 22, Spain with 22 and France with 22 visits. 588 visits came from a total of 30 countries. As it can be seen from the map, the website has users from across the world, which also shows the user's need and search for the SDA and GIS integration.

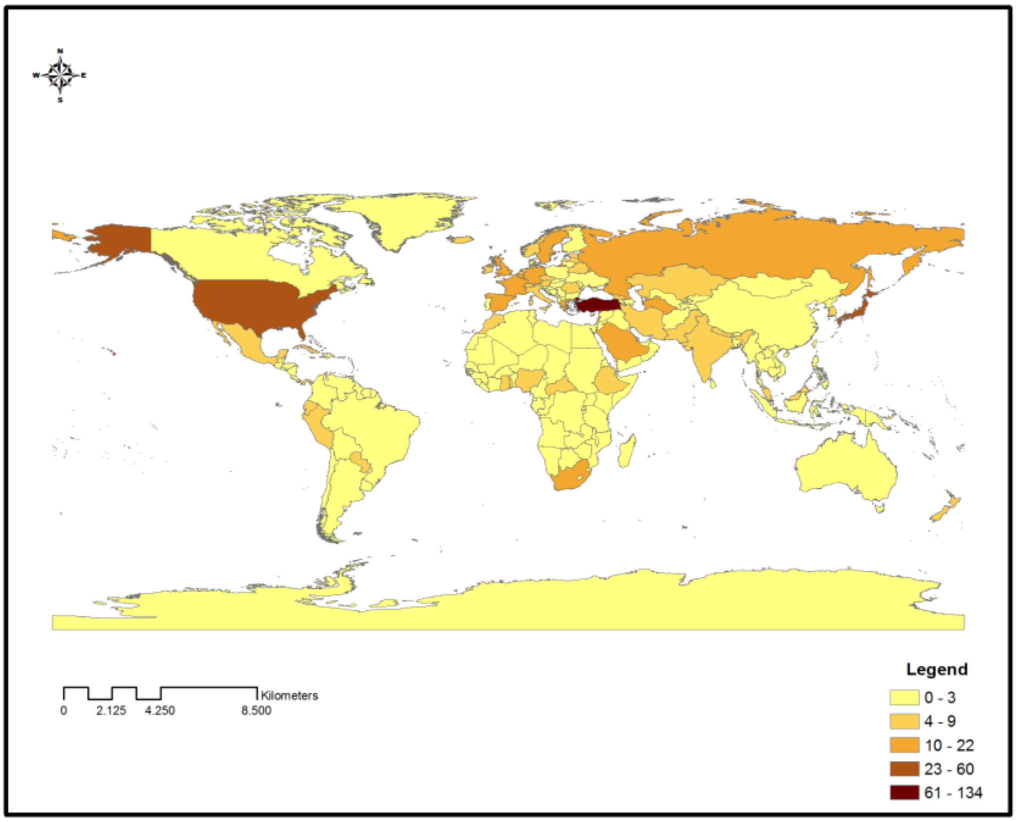


Figure 30. The Number of Visits Coming from Countries

Figure 31 shows the number of unique visitors who have visited the website more than 20 times across the world. 177 countries do not have any users who have visited the study website, while users from 30 countries have visited it at different times.

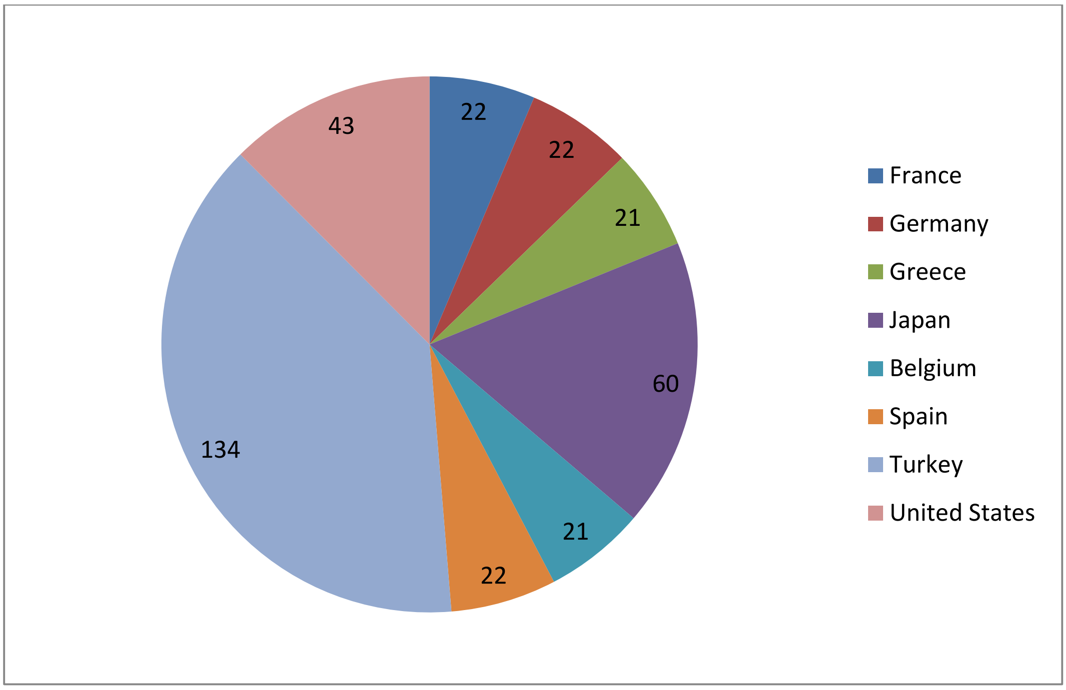


Figure 31. The Number of Visits across the Globe

In addition, it is interesting to analyse the user profile respect to their professions In this way, for further research, one can touch those people to provide a better solution to their specific problems.

A registration form is developed in the website. Users are required to fill a registration form in order to install the SDA system from the website. The Registered Users have filled the Survey Questionnaire after the software integration has been used.

Table 22 shows the numbers of the registered users and their countries.

Table 22. The Countries of the Registered Visitors

Name	Registered	Name	Registered
Azerbaijan	13	**Netherlands**	6
Australia	5	**Norway**	7
Bosnia and Herzegovina	14	**Qatar**	9
Brazil	17	**Russia**	14
Bulgaria	11	**Saudi Arabia**	16
Canada	16	**South Africa**	19
China	19	**Slovenia**	2
Ireland	8	**Tunisia**	2
Greece	21	**Turkey**	134
India	6	**Turkmenistan**	14
Iran (Islamic Republic of)	8	**United Kingdom**	15
Italy	9	**United States**	43
Japan	60	**Germany**	22
Lebanon	13	**Spain**	22
Belgium	21	**France**	22

Figure 32 shows the statistics about the users with a column graph. 99 sda4udig users are GIS experts; however, only two users are urban planners, which is the lowest number of professions among the sda4uDig users. To increase the usage and dissemination of the study, it is announced both on the uDig community and the R-sig-Geo platform. In this way, a satisfactory number of visitors has downloaded and completed the surveys in a short time.

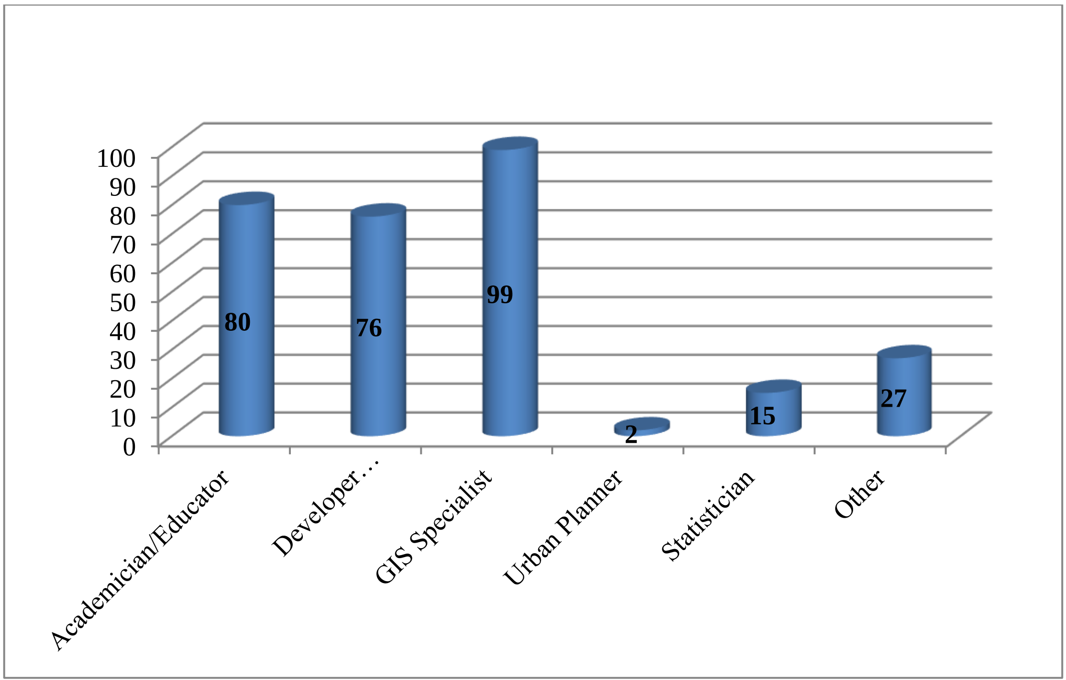


Figure 32. The Statistics of Visitors across the Globe with respect to their Professions

Finally, the statistics about the employment can be helpful to understand which part of the GIS community is most interested in such a study. Today, almost all the sectors are dealing with the GIS technology in one way or another. When these sectors are divided into different parts with respect to their usage levels, it is found out that governmental organizations use the GIS technology more heavily than the private sector. If a comparison is made among governmental organizations, it is seen that municipalities use the GIS tools and software mostly compared to the others (Cavur, Ozturan, Karaduman & Icli, 2015). However, Figure 33 shows the employment institutions of users with a column graph. It shows that users from the private sector and universities have shown the greatest interest in the study while those from governmental organizations are the least interested ones in this kind of software integration.

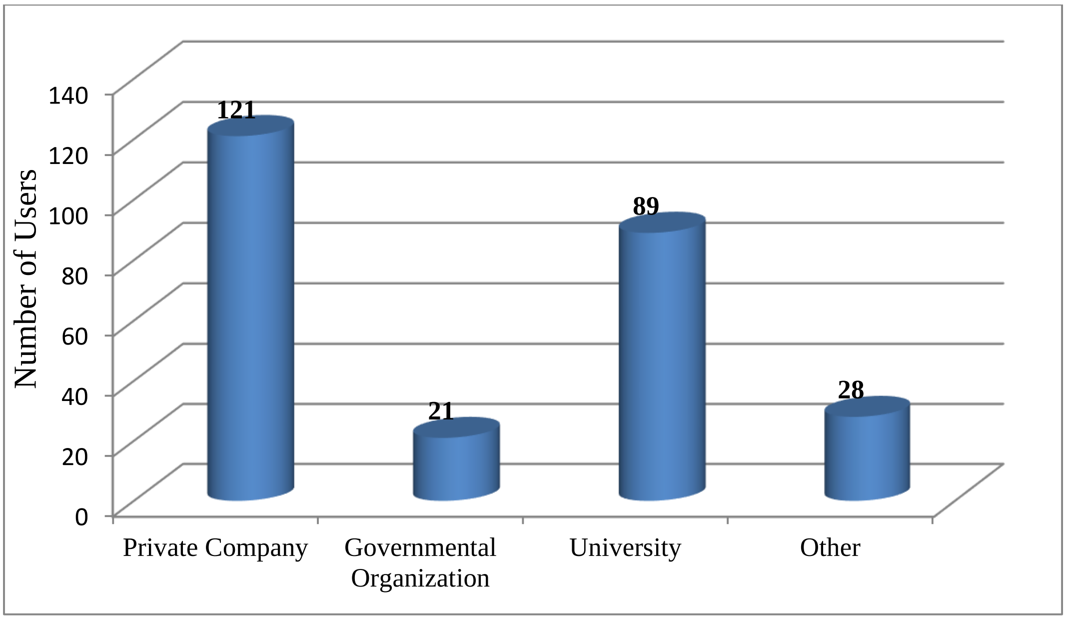


Figure 33. The Distribution of Website Visitors based on their Employment Institutions

Hands-on User Test Analysis:

To understand the user test result, descriptive statistics, ANOVA and linear regression tests are applied. Before explaining each test, their definition is given as follows.

Hypothesis 1

"Cronbach's α (alpha) is a coefficient of reliability. It is commonly used as a measure of the internal consistency and reliability. Theoretically, alpha varies from zero to 1, since it is the ratio of two variances" (Tavakol & Dennick, 2011).

A commonly accepted rule of thumb for describing internal consistency by using Cronbach's alpha is shown as $\alpha > 0.7$. To test each hypothesis, reliability analysis is conducted. After finding Cronbach's α (alpha) value for each piece of data, regression analysis is made for the data, whose Cronbach's α value is higher than 0.7. Table 23 shows the Cronbach's α (alpha) value for each piece of data. The hypothesis 1 is tested with the linear regression model.

Table 23. Cronbach's α (Alpha) Value for Hypothesis 1

Hypothesis Number	Hypothesis	Number of Item	Cronbach's α (Alpha)
1	Frequency Usage of GIS-SDA	32	0.78

The linear regression analysis is used to specify one-to-one relations between two or more different variables. The dependent variable of linear regression is “the success in diffusion of GIS” and the independent variable is “the frequency usage of GIS-SDA integration”. Linear regression analysis is used to test Hypothesis 1.

Table 24. Linear Regression Analysis of Hypothesis

Hypothesis Explanation	R^2	Sig.	B	Hypothesis	Hypothesis Acceptance
Frequency Usage of GIS-SDA	0.007	0.649	4.356	H1	**NOT Accepted**

It is claimed that there is a relationship between ***the frequency usages of SDA-GIS integration*** with the ***success level of GIS-SDA diffusion***.

Table 24 shows the result of regression analysis for Hypotheses 1. The frequency e of usage for the factor of GIS-SDA about GIS-SDA-related areas explains 1% of the variance in diffusion success, and has a positive impact by the proportion of 4.356 on GIS diffusion success.

According to the F statistics significance (Sig.) values in Table 24, all of the constructs are not meaningful because they are higher than 0.50, which means that these constructs can **NOT** be **accepted** as statistically meaningful with the confidence level of 95% and the GIS and SDA integration success cannot be explained by the related hypotheses. The R^2 values for the significant constructs show the percentage of the variance in the GIS-SDA project success, which is explained by the related hypotheses. B values indicate the path coefficients, which means that a 1-unit change in the related independent variables will affect the success of the GIS-SDA project by the proportion of B values.

Hypothesis 2

Table 25. ANOVA for the Satisfaction Levels of GIS-SDA Applications and the Quality of Integration

		Mean	F/t Value	Sig.
Hypothesis 2 (Accepted)	No Opinion	3.75	F=4.762	0.017
	Agree	4.22		
	Strongly Agree	4.80		
	Total	4.25		

Hypothesis 2 claims that there is a relationship between ***the satisfaction*** and ***the quality of GIS-SDA integration***. To test Hypothesis 2, ANOVA analysis has been conducted so that the success of the GIS-SDA integration is differentiated according to the satisfaction of integration and the quality of the integration. Table 25 describes the results of the ANOVA analysis that has been conducted in order to see whether the satisfaction levels of GIS-SDA differ in terms of the quality of integration.

It can be seen from Table 25 that the hypothesis is ***accepted*** since the significance value (.017) is lower than .05. It means that the quality of integration affects users differently in terms of the satisfaction levels of integration. In other words, the quality of integration is important for users in order to be satisfied while using this integrated software.

Hypothesis 3

Table 26. ANOVA for the Satisfaction Levels of GIS-SDA Applications and Documentation of the Integration

		Mean	F/t Value	Sig.
Hypothesis 3 NOT Accepted	Disagree	3.00	F=2.022	0.134
	No Opinion	4.28		
	Agree	4.26		
	Strongly Agree	4.50		
	Total	4.25		

Hypothesis 3 claims that there is a relationship between ***the satisfaction levels*** and the ***documentation of the GIS-SDA integration***. Table 26 describes the results of the ANOVA analysis that has been conducted in order to see whether the satisfaction levels of GIS-SDA differ in terms of the documentation of the integrated software. It can be seen from Table 26 that the hypothesis is ***NOT accepted*** since the significance value (.134) is higher than .05. It means that the documentation of integration does not affect users differently in terms of the satisfaction levels of integration. In other words, documentation is not important for users in order to be satisfied while using this integrated software.

Hypothesis 4

Table 27. ANOVA for the Satisfaction Levels of GIS-SDA Applications and Usability of Integration

		Mean	F/t Value	Sig.
Hypothesis 4 NOT Accepted	No Opinion	4.00	F=2.468	0.103
	Agree	4.14		
	Strongly Agree	4.62		
	Total	4.25		

Hypothesis 4 claims that there is a relationship between ***the satisfaction levels*** and ***the usability of the GIS-SDA integration***. Table 27 describes the results of the ANOVA analyses that have been conducted in order to see whether the satisfaction levels of GIS-SDA differ in terms of the usability of integration. It can be seen from Table 27 that the hypothesis is ***NOT accepted*** since the significance value (.103) is higher than .05. It means that the usability of integration does not affect users differently in terms of the satisfaction levels of integration. In other words, usability is not important for users in order to be satisfied while using the integrated software.

Hypothesis 5

Table 28. ANOVA for the Satisfaction Levels of GIS-SDA Applications and the Performance of Integration

		Mean	F/t Value	Sig.
Hypothesis 5 Accepted	Disagree	2.00	F=7.101	0.001
	No Opinion	4.07		
	Agree	4.30		
	Strongly Agree	4.75		
	Total	4.18		

Hypothesis 5 claims that there is a relationship between **the satisfaction** and ***the performance of the GIS-SDA integration***. Table 28 describes the results of the ANOVA analyses that have been conducted in order to see whether the satisfaction levels of GIS-SDA differ in terms of the performance of integration. It can be seen from Table 28 that the hypothesis is ***accepted*** since the significance value (.001) is lower than .05. It means that the performance of integration affects users differently in terms of the satisfaction levels of integration. In other words, performance is important for users in order to be satisfied while using this integrated software.

Hypothesis 6

Table 29. ANOVA for the Satisfaction Levels of GIS-SDA Applications and Ease of Integration

		Mean	F/t Value	Sig.
Hypothesis 6 Accepted	No Opinion	4.00	F=4.795	0.016
	Agree	3.87		
	Strongly Agree	4.57		
	Total	4.18		

Hypothesis 6 claims that there is a relationship between ***the satisfaction levels*** and ***the easiness of the GIS-SDA integration***. Table 29 describes the results of the ANOVA analyses that have been conducted in order to see whether the satisfaction levels of GIS-SDA differ in terms of the ease of integration.

It can be seen from Table 29 that the hypothesis is ***accepted*** since the significance value (.016) is lower than .05. It means that the ease of integration affects users differently in terms of the satisfaction levels of integration. In other words, the ease of integration is important for users in order to be satisfied while using this integrated software.

Hypothesis 7

Table 30. ANOVA for the Satisfaction Levels of GIS-SDA Applications and the Number of SDA Techniques of Integration

		Mean	**F/t Value**	**Sig.**
Hypothesis 7 Accepted	Strongly Disagree	2.00	F=10.917	0.000
	Disagree	5.00		
	No Opinion	3.80		
	Agree	4.33		
	Strongly Agree	4.75		
	Total	4.19		

Hypothesis 7 claims that there is a difference between users with a ***different number of geospatial techniques*** in terms of ***the satisfaction of the GIS-SDA integration***. Table 30 describes the results of the ANOVA analyses that have been conducted in order to see whether the satisfaction levels of GIS-SDA differ in terms of the number of SDA techniques of integration. It can be seen from Table 30 that the hypothesis is ***accepted*** since the significance value (.000) is lower than .05. It means that the number of SDA techniques of integration affects users differently in terms of the satisfaction levels of integration. In other words, the number of SDA techniques of integration is important for users in order to be satisfied while using this integrated software.

Hypothesis 8

Descriptive statistics are used to describe the basic features of the data in a study. They provide simple summaries about the sample and the measures. Together with simple graphics analysis, they form the basis of virtually every quantitative analysis of data (Socialresearchmethods.net, 2016). Table 31 shows the description of Hypothesis 8.

Table 31. Descriptive Statistics of Hypothesis 8

NO	Test Method	Hypothesis
H8	Descriptive	Visualization, SDA techniques, performance, GUI, effectiveness, utility and time are the main reasons for users to prefer and use the GIS software.

Hypothesis 8 claims that ***visualization, SDA techniques, performance, GUI, effectiveness, utility and time*** are the main reasons for users to prefer and use the GIS software. These hypotheses are just to show the descriptive statistics whether various criteria affect the usage of GIS-SDA integration or not. Since there are no applicable tests that can be applied, it does not make any decision about the acceptance or rejection of the hypothesis. Table 31 shows that 23 users accept that the SDA integration into GIS has brought the "Analysis Feature/Power" to the GIS software. On the other hand, only nine users have accepted that this integration increases the performance of GIS.

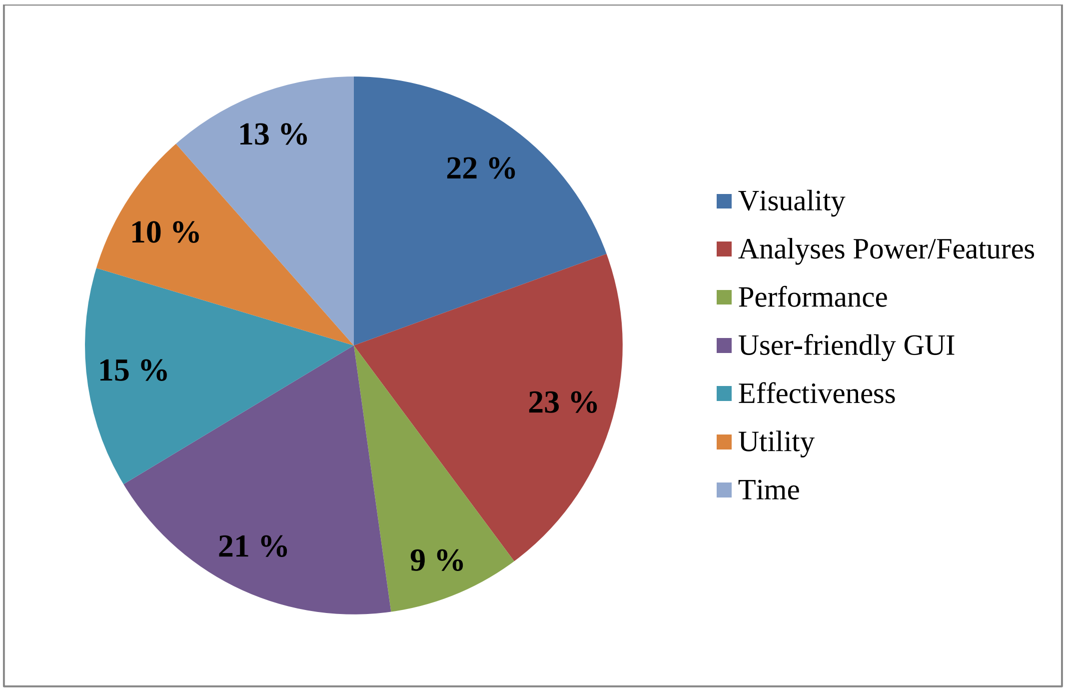


Figure 34. The Views of Users on the Advantages of sda4uDig over other SDA and GIS Software

Descriptive statistics, linear regression and ANOVA analyses are applied to test the user satisfaction and expectation from this study. Linear regression is applied to Hypotheses 1 and 9. There is no relationship between the usage frequency of such a software package and the satisfaction level with respect to Hypothesis 1. The result of the analysis proves that this hypothesis is not accepted. ANOVA test is applied on Hypotheses 2, 3, 4, 5, 6 and 7. The second hypothesis says that quality affects the satisfaction level of users. The analysis result shows that quality is important for users and the hypothesis is accepted. Hypothesis 3 claims that proper documentation for users increases the satisfaction level of them; however, the hypothesis is not accepted with respect to ANOVA test. Hypothesis 4 states that usability affects the satisfaction of users but the test result does not prove this hypothesis. Hypothesis 5 claims that performance increases the satisfaction of users. The test result supports this hypothesis. Hypothesis 5 claims that the ease of application is important to be satisfied with the usage of application. The ANOVA test result also proves that the ease is important for users. Hypothesis 7 says that the number of SDA techniques in GIS increases the satisfaction level of users. The test result supports Hypothesis 7 and it is accepted. Since Hypothesis 8 is tested with descriptive statistics, it cannot conclude that it is accepted or rejected. It shows some valuable statistics about several criteria. 72% of users believe that SDA enforces GIS with its analysis power but just 28% believes that SDA increases the performance of GIS.

In summary, the hypothesis constructed shows that there is a strong relation between SDA and GIS. Several criteria affect the success of integration and hence the satisfaction from this integration. Linear regression test shows that the hypothesis of “Frequency Usage of GIS-SDA” is not accepted. These tests show that “Quality of Integration”, “Performance of Integration”, “Easiness of Integration”, and “Number of SDA Techniques of Integration” are accepted. However, “Documentation of the Integration”, “Usability of Integration” are not accepted with respect to ANOVA tests. Finally, descriptive statistics show that 23 users among 32 believe that this study improves the “Analysis Power/Features” of GIS. All these hypotheses prove that one should take into account the quality of software, the performance of software, the ease of usage and the number of SDA techniques during the development of such an integration. On the other hand, the frequency of usage, documentation and the usability of integration is not important for such a study. All the analyses, statistics and tests are attached at the end of the thesis in Appendix E. This appendix contains figures from Figure E 1 to Figure E 7. It contains also tables from Table E 1 to Table E 31.

5.2. Installation of sda4uDig

The current versions were installed with an Internet connection at the beginning; this is because of the required R packages. Therefore, one can download the .exe file on www.sda4udig.com website and install the sda4udig without Internet connection. It means that it is installed in an online manner but it can operate offline. The details of installation is given in Appendix F. Appendix F includes figures from Figure F 1 to Figure F 17. However, the summary of the installation is explained in the following figures Figure 35 and Figure 36 show the prerequisite of the sda4uDig installation. The prerequisite of the installation is the version R 2.15.3. When the setup file is executed before the uDig GIS software installation, the R 2.15.3 version is installed.

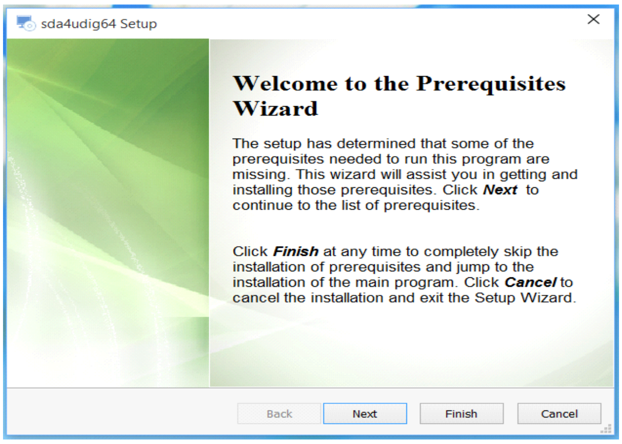


Figure 35. Welcome to the Prerequisites Wizard

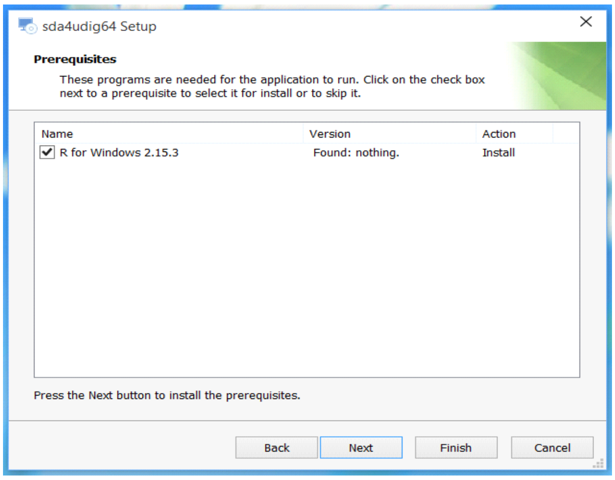


Figure 36. Prerequisite of Installation (R 2.15.3)

Figure 37 shows the first step of the sda4uDig GIS software and installation directory. The user just clicks the “Next” button to finalize the setup without any problems.

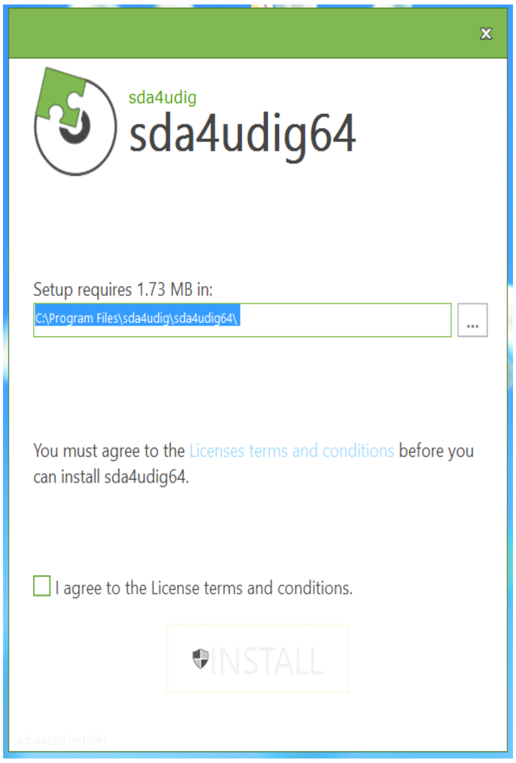


Figure 37. Installation of sda4uDig

After completing the installation, there is a menu to complete the installation including the R packages and some other setup options are as follows (Figure 38).

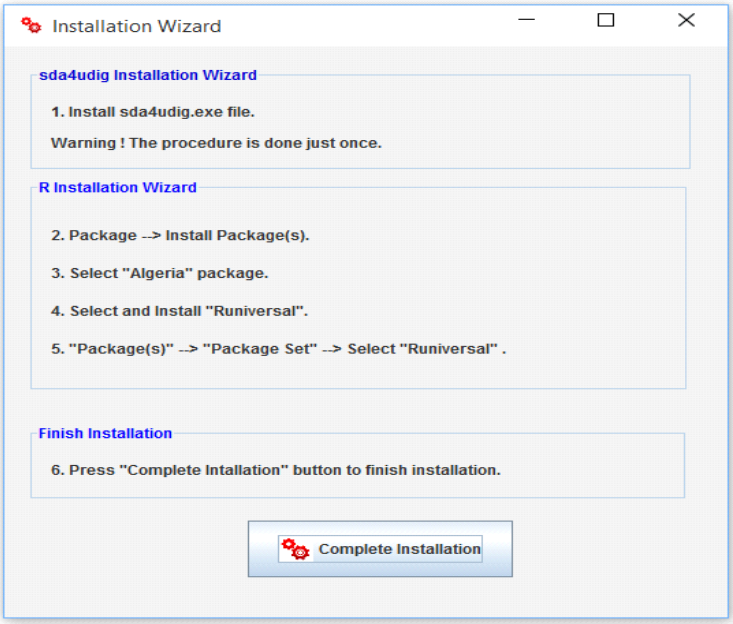


Figure 38. Complete Installation

In addition to the Windows sda4uDig software, the integrated software has been developed and tested for Linux OS. The interface of Linux version is as follows (Figure 39).

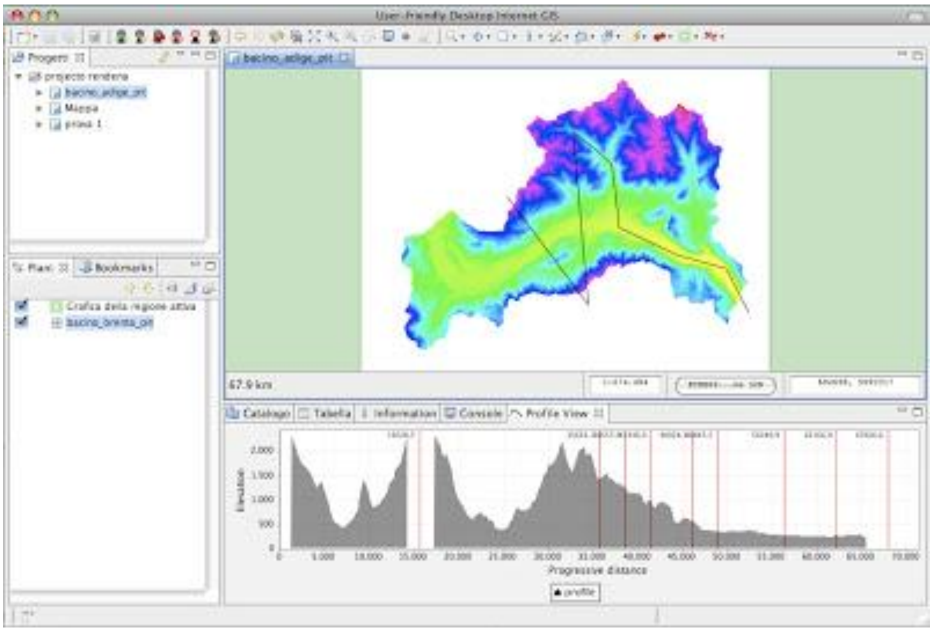


Figure 39. The Interface of sda4uDig on Linux OS

Although there are 14 geospatial methods and 4 non-geospatial methods integrated into the GIS software, just two case applications are applied in this part. The KDE technique is explained as a geospatial technique in this part and the column graph method is explained as a statistical technique. The proposed methodology is implemented for 14 different SDA methods, which are UI, KDE, ADE, G Function, L Function, F Function, K Function, KMS, Simulation envelope of CSR, FPM, Kriging, GWR, SR and Density Estimation, and 4 non-spatial functions, which are Histogram, Column Graph, Pie Chart and Statistical Summary tools. In addition, xls, csv and mdb import tools have been integrated into the GIS software but only xls import is applied in this part.

5.3. Application Examples

Three brief examples are presented to illustrate the uses of the interface. The data used is the accident data of Ankara collected in 2011. It covers an area of 24,521 km^2 and contains 4877 accident points distributed within the borders of the city of Ankara. The data contains the accidents that occurred on the roads of Ankara.

Each step of the proposed methodology was not explained and applied in this section. However, each step of the methodology, R Scripts, Java source code, all the geospatial and non-geospatial techniques applied, plug-ins, data and detailed documents about the sda4uDig can be found in the official website (http://www.sda4udig). In this section, the implementation of one geospatial technique is summarized with its interface, GUI and results. Appendix G shows the geospatial module integrated into the uDig GIS menubar. This appendix contains figures from Figure G 1 to Figure G 3. In this part, all the integrated techniques and their results are shown. In Figure 40, the interface of the uDig and data on some traffic accidents for Ankara are shown.

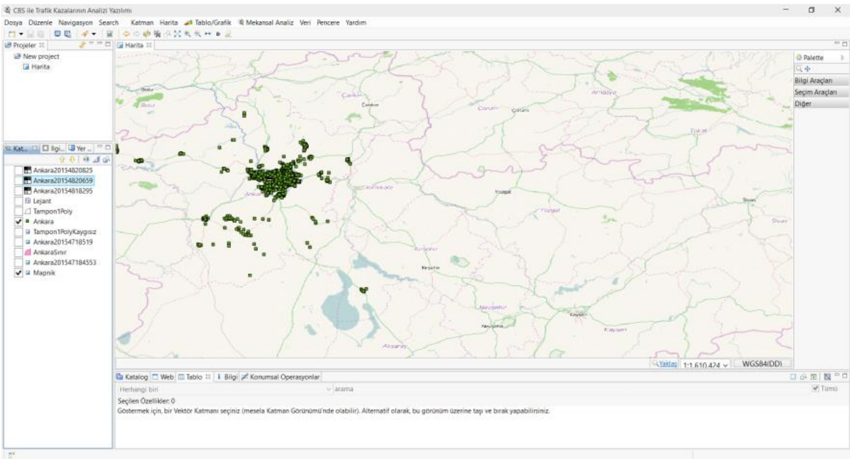


Figure 40. The uDig Interface with Traffic Accident Data

5.3.1. Kernel Density Estimation

The illustrated geospatial technique is the application of KDE on Ankara traffic accident data. The input, output and constraints of each method can be decided and defined with respect to requirements. Table 32 shows the input, output and other parameters of the KDE method. In regard to these arguments, the interface of that method is designed and created with Java programming language and the Window Builder on the Eclipse environment. The end user merely uses the interface of KDE and enters the input, output and other parameters as shown in Figure 41. The remaining geospatial, non-geospatial and integrated methods are attached at the end of the study as Appendix H. This appendix includes figures from Figure H 1 to Figure H 38.

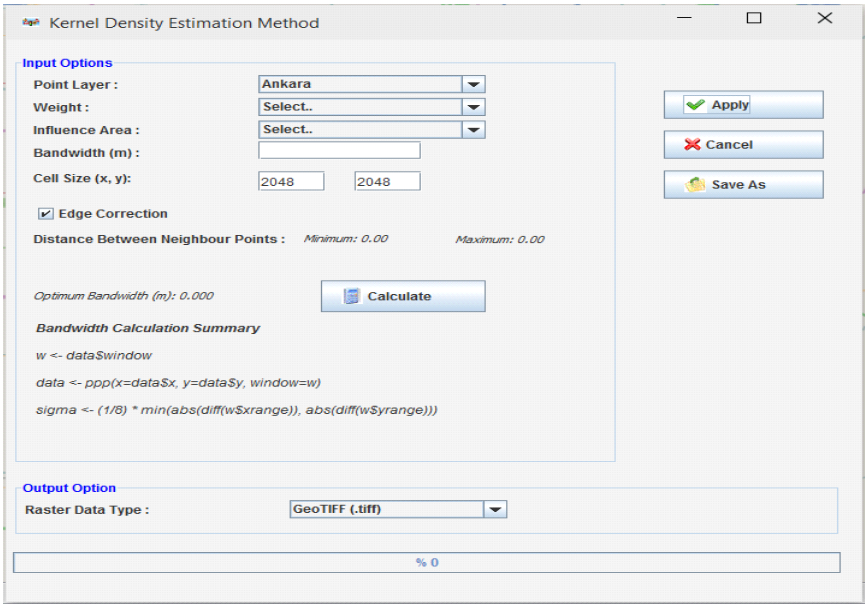


Figure 41. Interface of Kernel Density Estimation

Table 32. The Input, Output and Parameters of KDE Technique

I/O-Parameters	Explanation	Case Example
Input Layer	It accepts point (*.shp) data.	Ankara accident data.
Weight	Weight for selected column.	SONUCOLM: The dead people in accident.
Border	Border of the analysis result. Polygon *.shp file. Default is the border of point data.	Ankara Border.
Bandwidth	The bandwidth of the kernel is a free parameter which exhibits a strong influence on the resulting estimate.	3000 m
Cell Size	Number of cell in x and y directions.	X:2048; Y:2048
Edge Correction	Edge correction is applied if True.	True
Output Type	Jpeg, png, geotiff options are available.	GeoTiff
Calculate	It calculates the optimum bandwidth for users.	Not Applied.

Figure 42 shows the result of KDE on the uDig mapview. The color changes from cold (blue) to hot (red) in order to show riskier areas.

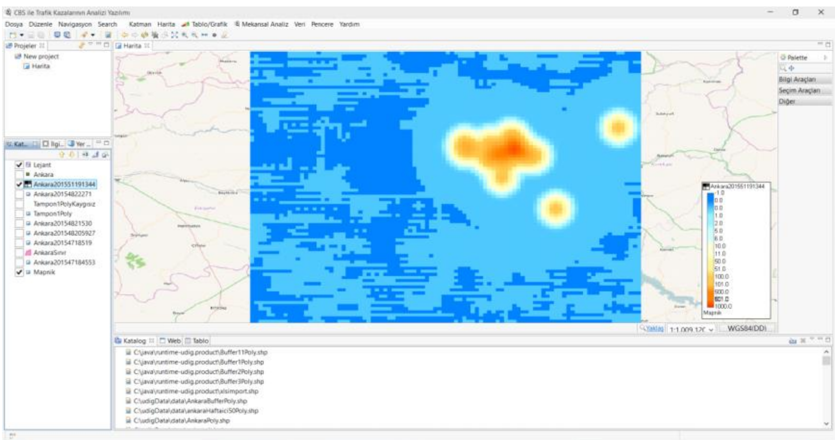


Figure 42. The Result of the KDE Technique Displayed on the uDig Mapview

5.3.2. Column Graph

In Figure 43, the input and output interface of Column Chart is shown.

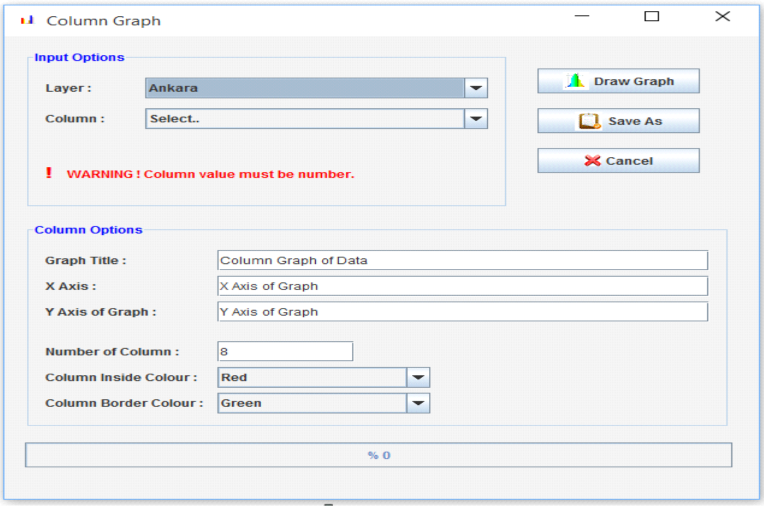


Figure 43. The Result of Column Graph

In Figure 44, the result of Column Graph using deadly accident data is shown. It shows the number of deaths in accidents with respect to weekdays. The first column shows that there are no deaths in 4853 accidents; however, there are 26 accidents with one dead person.

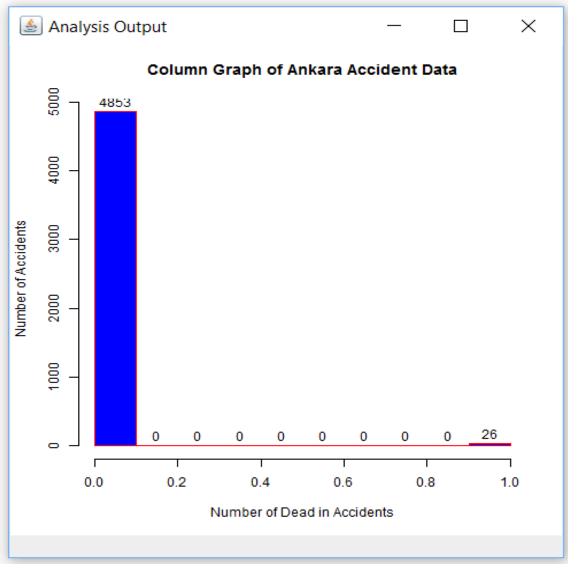


Figure 44. Column Graph Result

5.3.3. Import Excel File as a Shapefile into the Mapview of uDig

As pointed out before, the users have different types of data in hand and they want to use these data in the GIS software. In this part, there are three more different data types that have been adopted on the GIS software in addition to the shapefile. These data formats are ASCII, excel and mdb, which are shown in the following section.

In Figure 45, the input interface of Excel, Csv and Mdb is shown. All three input styles work in the same logic, and therefore only one of them is displayed.

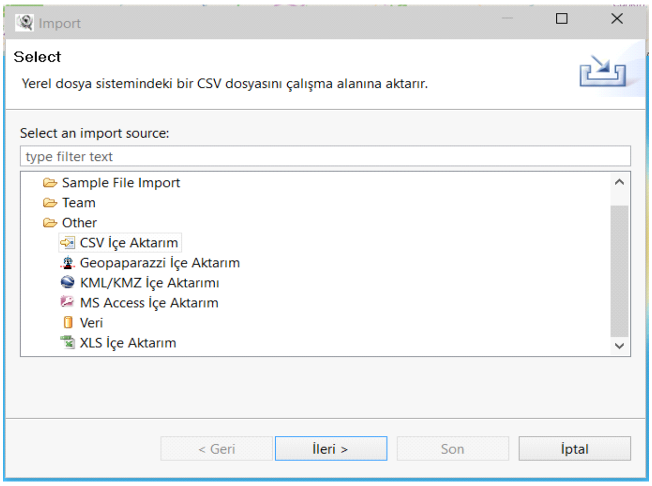


Figure 45. Excel, Csv and Mdb Input Import Interface

In Figure 46, users only select the x and y of data and then import the data in the GIS software as a shapefile. The metadata of other attributes can be selected as string, integer, long, float, text or can be left as default, which is string.

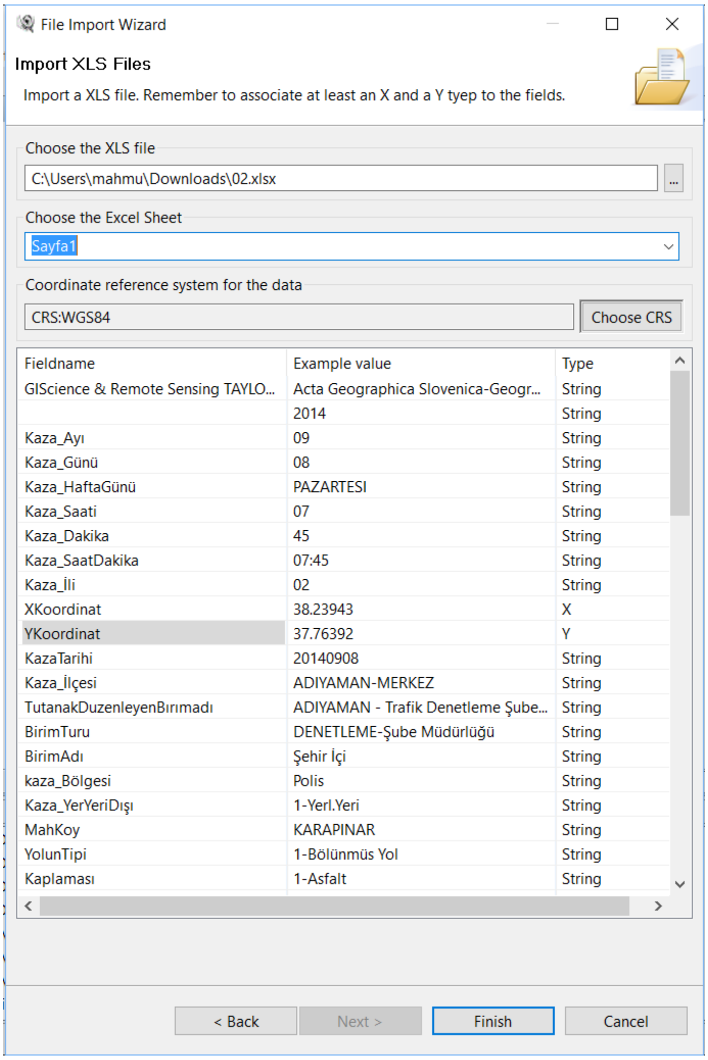


Figure 46. X and Y Coordinate Selection Interface

As before implied that there are various geospatial, non-geospatial techniques and several utility tools have been integrated into GIS software, three different techniques have been applied above in order to show the success of the study. Remaining techniques and tools are shown in Figure 47.

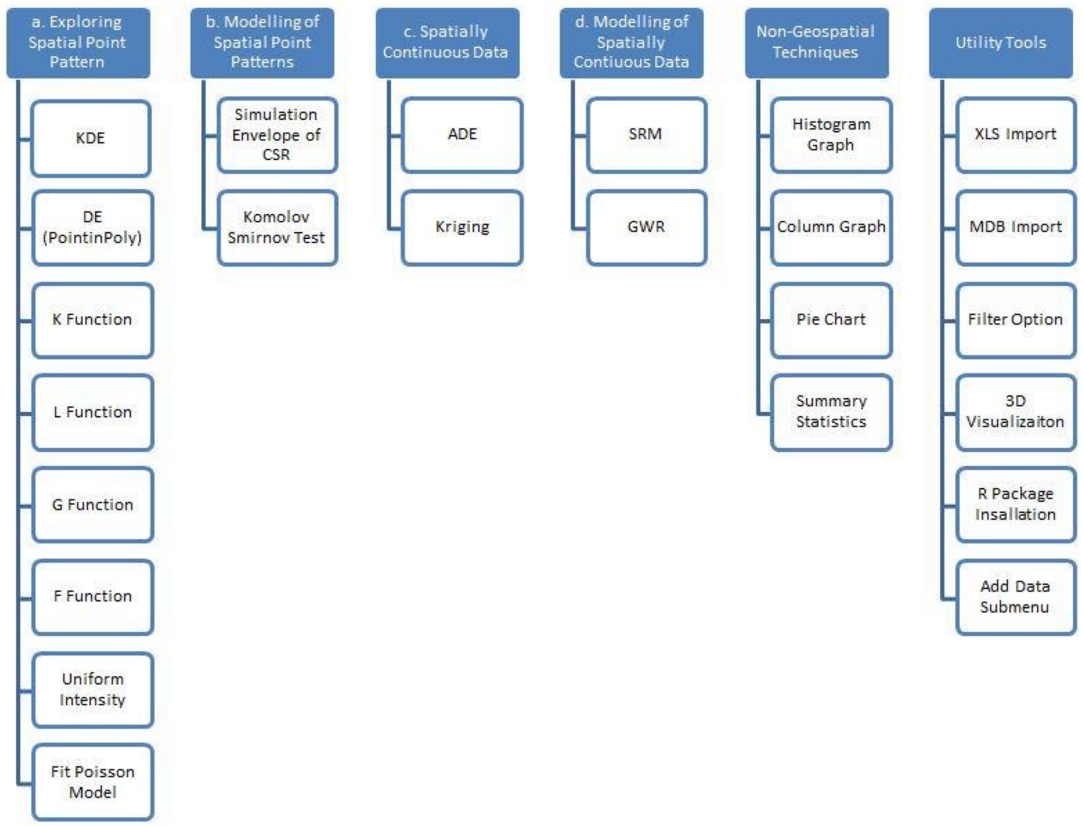


Figure 47. Geospatial, Non-geospatial and Utility Tools which are Integrated into GIS Software

The website for the study has been completed and a version of the sda4uDig has been distributed through this website. Some documents and source code of the study is also distributed through this website. The user can download the software and other source codes after filling a questionnaire. After using the sda4uDig software, one can fill another questionnaire to give feedback and some valuable insight about the study and for the support of the following versions of the sda4uDig. All the statistics about the user types, the country from which the sda4uDig is downloaded, and some more researches about this study will be reported later.

Figure 48 shows the interface of the website. The Appendix I contains Figure I 1.

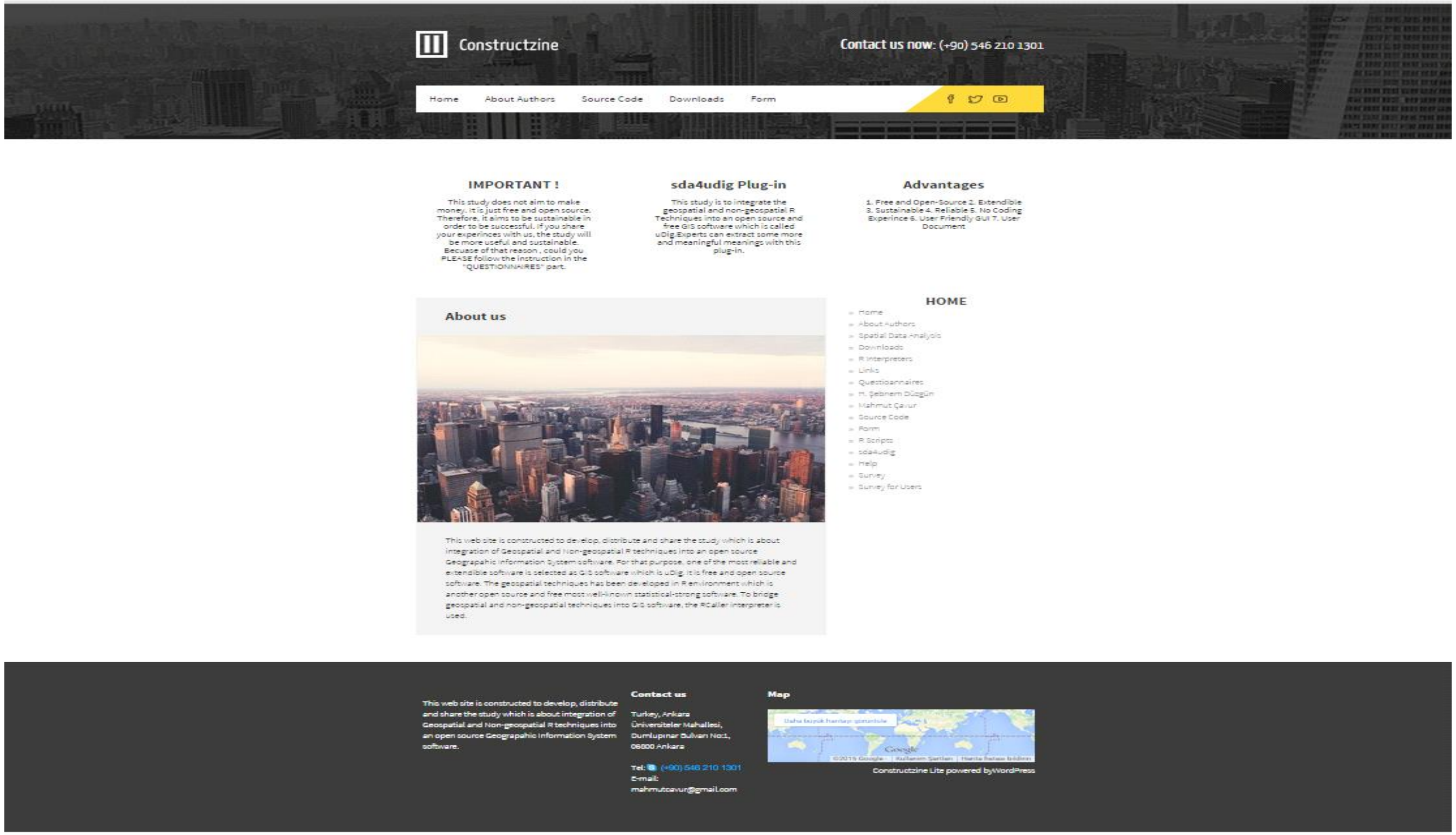


Figure 48. The sda4uDig Web-site (www.sda4udig.com)

5.4. Results and Discussion

Since the components of each step of the framework are compatible with each other and also the components work successfully, the proposed framework is deemed successful on the whole. The application examples including the installation of the sda4uDig prove that the integrated software is operational by using the framework and methodology. As the application examples clearly define each step of usage (see Chapter 5), the user can enter input data into a user-friendly interface and get the result of the applied technique in a very short time. The success of these examples indicates that the study is successful in regard to the ***user's perspective***. In Chapter 5, the application of the proposed framework is tested with a case study. Each step of this proposed framework is proven by this case study, whereas the methodology is tested and proven by a column graph example. The case study and the example prove that the proposed framework successfully works in regard to the ***developer's perspective***.

To test the performance, usability and several other criteria with respect to the developers and users, a test methodology was developed. This methodology contained both the developer test and the user test. The ***developer test*** was applied to each step of the coding phase and iterated until successful implementation was achieved. Therefore, the study could not proceed until success was achieved after the developer test. However, proceeding to the next step, i.e. the user test, indicates the success of the developer test.

Two surveys were applied to the users in order to collect data and their feedback on the ***user test***. The collected data was analyzed. To get feedback and their perspective, an online website (Figure 48) was developed and shared with the uDig community and the R community. This way increased the usage, distribution and dissemination of the study. 558 visits from 30 countries in six months infer that there is a demand for such a study. These numbers prove that the study will be of great interest in the future. Furthermore, the registered visitors (Figure 32) are mostly the users who are experts in GIS and SDA. 99 users over 299 registered visitors are GIS specialists and 76 of them are developers. The number of these developers promises that the study can be supported by developers in the future. In Figure 33, the 121 registered users are from the private sector and 89 of them are from universities; however, the interest from governmental organization is 21 over 259 visitors. It shows that there is a demand from private and academic environment for such a study, but currently it is not of interest within the governmental organizations.

In addition to the online test, the hands-on test was applied to 32 GIS and SDA users at the GGIT at METU. Before the application of the hands-on test, several hypotheses were suggested to test the success of the sda4uDig. With respect to these hypotheses, a hands-on questionnaire was prepared for users. After that, 32 users were given a user task (Appendix D) to follow and apply on the sda4uDig integration. They were also given a questionnaire survey (Appendix C) to get their feedback, insight and experience on the sda4uDig after the user task. Then, all these data were analyzed with the SPSS 15 statistical tool to test the hypotheses with respect to the collected data. As expected, several of the hypotheses were accepted and several of them were rejected. The descriptive statistics, reliability test, linear regression and Anova test were applied to test each hypothesis. Hypothesis 1 indicates that the usage frequency does not affect the diffusion success of such a study. Hence, this hypothesis is rejected. Likewise, the documentation of such a study (Hypothesis 3) is not a good indicator to prove the success of the study and the satisfaction of users. The most interesting result is obtained from Hypothesis 4. According to this, the usability is not accepted as a good indicator for the success of the study and the satisfaction level of users. This is one of the unexpected results from the hypotheses since 21% of the users claim that the user-friendly GUI of the software is important for them, based on the results of the descriptive statistics in Figure 34. However, the results of Anova for Hypothesis 4 (Table 27) contradict with this result.

Although usage frequency and documentation of the study is statistically rejected, there are several other criteria proving the success of this study. The acceptance of the quality of the study (Hypothesis 2), the performance of the study (Hypothesis 5), the ease of the integrated software (Hypothesis 6) and the number of SDA techniques integrated into GIS (Hypothesis 7) show that there is a relation between the satisfaction levels of users and the success of the study. All these hypotheses were tested with Anova analysis and accepted. Likewise, in Hypothesis 5, users claim that the performance-oriented software is important for them, being similar to the results given in Figure 34. Similarly, users claim that the ease of the integrated software is very important for them in Hypothesis 6. Since 22% of the users claim that the visuality and 21% of them claim that the user-friendly GUI is important in Figure 34, this hypothesis also meets the expectation of users. Finally, the numbers of SDA techniques are regarded as important for the users with respect to Hypothesis 7. Since the number and type of SDA techniques in GIS enforce the power of GIS, the result is logical and expected. Besides, in Figure 34, 23% of the users accept that the SDA techniques in GIS are critical for them.

Consequently, it can be said that the framework and the methodology work successfully. Their success is proven with the case studies and examples. The problems of users have been solved with the user-friendly integrated software. Also, the framework and the methodology are provided for developers to integrate any required SDA technique into any GIS software easily and methodologically. The test methodology is applied to the users to discover whether the results are consistent or not. The developer test is successfully achieved. The user test is achieved and their results are analyzed methodologically. For this purpose, the hypothesis is conducted, the user task is prepared and analyzed with the SPSS tools. The results of analyses are satisfactorily coherent with the expectations. The result of the hypothesis proves that there is a strong relation between SDA and GIS and this shows that conducting such a study is valuable for the scientific world. The number of visitors from the world even for a limited time is actually important and shows the interest of the GIS and SDA communities in such a study.

CHAPTER 6

CONCLUSION AND FURTHER RECOMMENDATIONS

6.1.Conclusion

The main aim of the study is to provide a framework for developers to embed any required spatial statistical and non-spatial technique(s) into a GIS software methodologically with suitable software including SDA and GIS, bridge technique, coupling strategy and other selected parts and compatible components. The uDig and the R are two software types which are integrated. R is a free and open-source spatial statistical, non-spatial and image data analysis software. It is one of the most acceptable and most widely used software packages in the world. But it is not user-friendly, and can only be coded by developers and the third parties depending on using the RScript. Each development is included in the R-package and shared with a detailed document. The uDig is also an open-source GIS software program that can run on all the OS's. It can be extended with a plug-in, and is also compatible with the R software. It is developed by an object-oriented language called the Java programming language. Similar to the uDig, the RCaller is developed by Java programming language and it is open-source.

With the selected components, a straightforward framework is proposed for the developers. Each step of the framework is constructed with respect to the components of GIS. A developer can follow and apply each step of the framework in order to select, evaluate and embed spatial statistical techniques into a GIS software easily. Framework is not only provided for the selected components of this study but is also applicable for any selected spatial statistical technique and GIS software including the bridge technique. The only discussion can be made about the selection of the coupling strategy, which is tight coupling strategy. Instead of tight coupling strategy, one can use full coupling methodology, which is the best strategy if the selected components are suitable to apply this strategy; yet the uDig does not contain any macro language and do not allow to use full coupling strategy.

All in all, the framework is successfully implemented for this study without any problems. The methodology explains each step of the application of the framework. A developer can use this methodology to embed any technique into the GIS software. It is easy to understand and be applied by mid-level developers as well.

Three case studies, Kernel Density Estimation, Column Graph and xls Import, are applied and explained with their input, GUI and results. The application of the framework and the methodology with these three case studies proves the success of the study indeed.

Finally, in order to understand users' idea and get feedback from them, a survey questionnaire is applied to them. This is achieved in two ways: one is to develop a web page for this study and distribute the source code, documents and the questionnaire surveys; the second is to apply the survey questionnaire to the users in a hands-on manner. To do so, the students of two lectures related to GIS at the GGIT at METU were selected for hands-on user test and survey. The surveys have been analyzed and the result is reported. 33 % of the registered users are GIS specialists which are the most interested group. 134 visitors of web site are from Turkey and 60 of them are from Japan which are the most interested countries to this study. Universities (89 visitors) and private companies (121 visitors) have downloaded the study over 259 visitors since it was published in the web. In addition, several hypotheses which measure the success of several criteria and the success of GIS are tested. The results of these hypotheses are also reported in this study. "Usage Frequency", "Documentation about the Study", "Usability of the Study" are found to be not significant factors to indicate the success of the integration with respect to hypotheses. On the other, "Quality of the Integration", "Performance of the Integration", Easiness of the Usage of the Integration" and "Number of SDA Techniques" are obtained to be the key factors to affect the success of the study with respect to hypotheses. Lastly, 23% of the users claim that SDA add “Analysis Power/Feature” to the GIS and 22 % claim that SDA increase the visualization feature of GIS softwarewith respect to descriptive statistics. According to these surveys, the users have been satisfied with the result of this study mostly.

6.2.Further Recommendations

Further, there is great need to carry out further research on how to develop graphics in the uDig that are more suitable for statistical purposes, such as advanced geospatial techniques and interactive geospatial techniques and how to enhance the dynamic and interactive linking between a mapview, chart and table in the uDig. The effort to develop a statistical module in GIS directly, however, should not be hampered by these minor limitations. Our example, by focusing on how to integrate standard geospatial analysis and the test of spatial technique on the uDig, exemplifies how GIS and statistical analysis can benefit from each other. A statistical module in GIS can take full advantage of the topology relationships and the visualization tools provided by GIS. GIS, on the other hand, may use this statistical module to extend its analytical capabilities from simple descriptive statistics to more exploratory and inferential data analysis. This example of implementation, however, is far from covering all the statistical methods that can potentially be integrated into GIS. We discussed how to implement other local spatial statistics, such as G Function and F Function, which is based on the distance connectivity matrix, not the spatial autoregressive models. Although there is a general consensus that GIS should increase its abilities on statistical analyses and develop its own statistical module, which techniques should be included in this module is still open to debate. Obviously, more interesting questions and challenges will emerge from the area of integrating GIS with statistical analysis. The outcomes reported in this paper may establish a foundation for us to develop further SDA functions in the uDig with R. Future work will cover the use of other R interpreters to improve the efficiency and performance of the integration. Furthermore, geospatial techniques can be developed in the web-based manner to increase the spread of GIS in the near future.

In addition, there are so many image processing packages in R that this can trigger researchers to design, develop and improve the capabilities of GIS with an image processing toolbox.

Finally, the data used with the uDig GIS is limited in size and currently there is a big demand to use and analyze big data to understand the geospatial data. Providing a big data analytics for users really attracts the attention of the GIS and geospatial community. Therefore, the next step after this study will be conducting a study on big geospatial data analytics by the owners of this study.

Code availability

uDig is available at http://udig.refractions.net/ and https://github.com/udig

R is available at http://www.r-project.org/

sda4udig stable components and code documentation are available on http://sda4udig.com/

There, for each component are presented input, output, parameters description, a descriptive help and a test case.

REFERENCES

A Cinnober white paper, (2012). The benefits of using Java as a high-performance language for mission critical financial applications.

Ai-geostats.org, (2015). SoftwareGeostatistical < AI_GEOSTATS < TWiki. [online] Available at: http://www.ai-geostats.org/bin/view/AI_GEOSTATS/SoftwareGeostatistical [Accessed 23 Feb. 2015].

Akbari, M., & Rajabi, M. (2013). Evaluation of Desktop Free/Open Source GIS Software Based on Functional and Non-Functional Capabilities. Search.ebscohost.com. Retrieved 16 March 2016, from http://search.ebscohost.com/login.aspx?direct=true&db=a9h&AN=91761951&site=ehost-livenhttp://content.ebscohost.com/ContentServer.asp?T=P&P=AN&K=91761951&S=R&D=a9h&EbscoContent=dGJyMNXb4kSeqLQ4xNvgOLCmr0yep7BSrqi4SraWxWXS&ContentCustomer=dGJyMPGotVGxrrNPuePfgeyx44Dt6fIA

Anon. (2016). Definition of METHODOLOGY. [online] Merriam-webster.com. Available at: <http://www.merriam-webster.com/dictionary/methodology> [Accessed 15 Mar. 2016].

Anselin, L. (1992). Spatial Data Analysis Wıth Gis : An Introduction To Appiıcatıon in The Social Sciences (pp. 92-100).

Anselin, L. (2000). Computing environments for spatial data analysis. Journal Of Geographical Systems, 2(3), 201-220. http://dx.doi.org/10.1007/pl00011455

Anselin, L., & Bao, S. (1997). Exploratory spatial data analysis linking SpaceStat and ArcView. [S.l.]: [s.n.].

Anselin, L., & Getis, A. (1992). Spatial statistical analysis and geographic information systems. Ann Reg Sci, 26(1), 19-33. http://dx.doi.org/10.1007/bf01581478

Anselin, L., Dodson, R. F., & Hudak, S. (1993). Linking GIS and spatial data analysis in practice. In *Geographical Systems.* (1 ed., Vol. 1, pp. 3-23)

Anselin, L., Syabri, I., & Kho, Y. (2006). GeoDa: An Introduction to Spatial Data Analysis. Geographical Analysis, 38(1), 5-22. http://dx.doi.org/10.1111/j.0016-7363.2005.00671.x

Bailey, T.C. & Gatrell, A.C. (1995). Interactive spatial data analysis Longman Higher Education, Harlow

Batty, M., & Xie, Y. (1994). From cells to cities. Environment And Planning B: Planning And Design, 21(7), s31-s48. http://dx.doi.org/10.1068/b21s031

Bhatt, G., Kumar, M., & Duffy, C. (2014). A tightly coupled GIS and distributed hydrologic modeling framework. Environmental Modelling & Software, 62, 70-84. http://dx.doi.org/10.1016/j.envsoft.2014.08.003

Bonin, O. (2012). How to make R, PostGIS and QGis cooperate for statistical modelling duties: a case study on hedonic regressions. In . In: OGRS Open Source Geospatial Research & Education Symposium..

Brandmeyer, J., & Karimi, H. (2000). Coupling methodologies for environmental models. Environmental Modelling & Software, 15(5), 479-488. http://dx.doi.org/10.1016/s1364-8152(00)00027-x

Burback, R. (1998). Software Engineering Methodology : The Watersluice c Copyright 1999.

Cascadoss, (2007), Business Models, CASCADOSS, Development of Transnational Cascade Program on Open Source GIS & RS Software for Environmental Applications, Sixth Framework Program, Identification of New Methods of Promoting and Encouraging Transnational Technology Transfer, http://www.cascadoss.eu/en/index.php?option=com_content&task=view&id=60&Ite mid=67, last accessed on August 11, 2010.

Cavur, M., Özturan, M., Karaduman, C., & Içli, A. B. (2015). "Diffusion of GIS At Municipalities In Istanbul". In IMCET, April 14-17 2015.

Chambers, J. M. (2008). Software for Data Analysis: Programming with R. Biometrics, 65(4), 1313-1313. doi:10.1111/j.1541-0420.2009.01343_10.x

Confino, J., & Laplante, P. (2010). An Open Source Software Evaluation Model. International Journal Of Strategic Information Technology And Applications, 1(1), 60-77.

Cook, D., Majure, J.J., and Symanzik, J. (1996). Dynamic Graphics in a GIS : A Bidirectional Link between ArcView 2 . 1 TM and XGobi | An Update 1 Introduction 2 Integration of Interactive and Dynamic Graphics Tools into a GIS.

Cook, D., Symanzik, J., Majure, J., & Cressie, N. (1997). Dynamic graphics in a GIS: More examples using linked software. Computers & Geosciences, 23(4), 371-385. http://dx.doi.org/10.1016/s0098-3004(97)00015-0

Cremers, N., Wouters, W., Roo, A., Wesseling, C., Burrough, P., & Van Deursen, W. (1995). Integratie GIS en Dynamic Modellen "Definitiestudie" (p. 34). Survey Department, Directorate-General for Public Works and Water Management (MD-GAG-NR.9503).

Definition of Methodology. (2016). Merriam-webster.com. Retrieved 15 March 2016, from http://www.merriam-webster.com/dictionary/methodology

DeVantier, B.A., Feldman, A.D., 1993. Review of GIS applications in hydrologic modeling. J. Water Resour. Plan. Manag. 119, 246e261.

Duckham, M., M.F. Goodchild, and M.F. Worboys, editors, 2003. Foundations of Geographic Information Science. New York: Taylor and Francis.

Dueker, K.J. and D. Kjerne. (1989). Multipurpose cadastre: Terms and definitions. Am. Soc. Photogra. Remote Sens. and Am. Congr. Surv. Mapping, Falls Church, Va. 5:94-103.

Duzgun, H.S. (2012). Spatial Data Analysis. Presentation, METU.

Everitt, B., & Hothorn, T. (2006). A handbook of statistical analyses using R. Boca Raton: Chapman & Hall/CRC.

Fischer, M. M., & Nijkamp, P. (1992). Geographic information systems and spatial analysis. The Annals of Regional Science, 26(1), 3–17. http://doi.org/10.1007/BF01581477

Fotheringham, A. & Rogerson, P. (1993). Research Directions in GIS and Spatial Analysis. London: Taylor and Francis

Fotheringham, A., & Rogerson, P. (1994). Spatial analysis and GIS. London: Taylor & Francis.

Goodchild, M. (1987). A spatial analytical perspective on geographical information systems.International Journal Of Geographical Information Systems, 1(4), 327-334. http://dx.doi.org/10.1080/02693798708927820

Goodchild, M. (1988). Stepping Over The Line: Technological Constraints And the New Cartography.Cartography And Geographic Information Science, 15(3), 311-319. http://dx.doi.org/10.1559/152304088783886973

Goodchild, M. F., & Longley, P. (1999). The future of GIS and spatial analysis. Geographical Information Systems, 1, 567–580. Retrieved from http://www.geos.ed.ac.uk/~gisteac/gis_book_abridged/files/ch40.pdf

Goodchild, M., & Haining, R. (2003). GIS and spatial data analysis: Converging perspectives. Papers In Regional Science, 83(1), 363-385. http://dx.doi.org/10.1007/s10110-003-0190-y

Goodchild, M., Anselin, L., Appelbaum, R., & Harthorn, B. (2000). Toward Spatially Integrated Social Science. International Regional Science Review, 23(2), 139-159. http://dx.doi.org/10.1177/016001760002300201

Goodchild, M., Haining, R., & Wise, S. (1992). Integrating GIS and spatial data analysis: problems and possibilities. International Journal Of Geographical Information Systems, 6(5), 407-423. http://dx.doi.org/10.1080/02693799208901923

GRASS GIS, (2008a), GRASS: Introduction: What‟s GRASS, Geographic Resources Analysis Support System, http://www.grass.itc.it/intro/general.php, last accessed on September 5, 2008

GRASS GIS, (2008b), GRASS: Introduction: First time Users of GRASS, Geographic Resources Analysis Support System, http://www.grass.itc.it/intro/firsttime.php, last accessed on September 5, 2008

Griffith, D. (1993). Advanced spatial statistics for analysing and visualizing geo-referenced data.International Journal Of Geographical Information Systems, 7(2), 107-123. http://dx.doi.org/10.1080/02693799308901945

Gurbani, V., Garvert, A., & Herbsleb, J. (2005). A case study of open source tools and practices in a commercial setting. SIGSOFT Softw. Eng. Notes, 30(4), 1. http://dx.doi.org/10.1145/1082983.1083264

Haining, R. (1989). Geography and Spatial Statistics: Current Positions, Future Developments (pp. 191-203). Oxford: Basil Blackwell.

Hégron, G., Bocher, E., & Petit, G. (2012). The open source GIS, an ideal framework for the development of an integrated modelling platform devoted to sustainable urban planning: first steps with OrbisGIS and CartoPolis. InOGRS Open Source Geospatial Research & Education Symposium (pp. 59-65). Yverdon-les-Bains: OGRS.

Henning, E. (2004). Finding your way in qualitative research. Pretoria: van Schaik.

Heywood, I. (1990). GIS in the Social Sciences. Environment and Planning. A.22:849-854.

Karimi, H., & Houston, B. (1996). Evaluating strategies for integrating environmental models with GIS: Current trends and future needs. Computers, Environment And Urban Systems, 20(6), 413-425. http://dx.doi.org/10.1016/s0198-9715(97)00006-9

Kemec, S., & Duzgun, H.S. (2006). 3D Visualization for Urban Earthquake Risk. In Conference: Geohazards - Technical, Economical and Social Risk Evaluation (pp. 37-47).

Kepoglu, V. (2011). Development Of Free/Libre and Open Source Spatial Data Analysis System Fully Coupled with Geographic Information System (Unpublished doctoral thesis). Middle East Technical University Ankara, Turkey.

Levine, N. (2006). Crime Mapping and the Crimestat Program. Geographical Analysis, 38(1), 41-56. http://dx.doi.org/10.1111/j.0016-7363.2005.00673.x

Longley, P. (2001). Geographic information systems and science. Chichester: Wiley.

Maindonald, J., & Braun, W. (2006). Data Analysis and Graphics Using R An Example-based Approach.

Massei, G., Rocchi, L., Paolotti, L., Greco, S., & Boggia, A. (2012). MCDA-GIS integration: an application in GRASS GIS 6.4. In OGRS Open Source Geospatial Research & Education Symposium. (pp. 195-201).

Mikelbank, B. (2010). Quantitative Geography: Perspectives on Spatial Data Analysis, by A. S. Fotheringham, C. Brunsdon, and M. Charlton. Geographical Analysis, 33(4), 370-372. http://dx.doi.org/10.1111/j.1538-4632.2001.tb00453.x

Mitasova, H., & Neteler, M. (2004). GRASS as Open Source Free Software GIS: Accomplishments and Perspectives. Transactions In GIS, 8(2), 145-154. http://dx.doi.org/10.1111/j.1467-9671.2004.00172.x

Mouton, J. (1996). Understanding social research. Pretoria: Van Schaik Publishers.

Munassar, N., & Govardhan, A. (2010). Comparison between five models of software engineering. International Journal of Computer Science Issues, 7(5), 94–101.

Murray, A. T. (2010). Quantitative Geography. Journal of Regional Science, 50(1), 143–163. http://doi.org/10.1111/j.1467-9787.2009.00642.x

Neteler, M., Beaudette, D. E., Cavallini, P., Lami, L., and Cepicky, J., (2008), GRASS GIS, in Open Source Approaches in Spatial Data Handling, ed. by G. Brent Hall and Michael G. Leahy, Springer, Berlin.

Palmer, M., Bailey, T. and Gatrell, A. (1996). Interactive Spatial Data Analysis. Ecology, 77(5), p.1642.

Pruim, R. (2016). Computational Statistics using R and RStudio. Presentation, Calvin College.

Raju, P. L. N. (2003). Spatial Data Analysis. Satellite Remote Sensing and GIS Applications in Agricultural Meteorology, 151–174.

Renjin.org | The JVM-based interpreter for the R language for statistical computing. (2015). Renjin.org. Retrieved 8 February 2015, from http://www.renjin.org/

Rey, S. (2012). Open source spatial analysis: lessons for research and education from PySAL. In OGRS Open Source Geospatial Research & Education Symposium (pp. 66-71).

RServe - Introduction and Installation. (2015). Studytrails.com. Retrieved 15 February 2015, from http://www.studytrails.com/R/RServe/RServe-Introduction-Installation.jsp

Satman, M. (2014). RCaller: A Software Library for Calling R from Java. BJMCS, 4(15), 2188-2196. http://dx.doi.org/10.9734/bjmcs/2014/10902

Spatial data analysis. (2016). TheFreeDictionary.com. Retrieved 9 January 2016, from http://encyclopedia2.thefreedictionary.com/Spatial+data+analysis

Star, J., & Estes, J. (1991). Geographic information systems: An introduction. Geocarto International, 6(1), 46-46. http://dx.doi.org/10.1080/10106049109354297

Steiniger, S., & Bocher, E. (2009). An overview on current free and open source desktop GIS developments. International Journal Of Geographical Information Science, 23(10), 1345-1370. http://dx.doi.org/10.1080/13658810802634956

Sui, D., & Maggio, R. (1999). Integrating GIS with hydrological modeling: practices, problems, and prospects. Computers, Environment And Urban Systems, 23(1), 33-51. http://dx.doi.org/10.1016/s0198-9715(98)00052-0

Symanzik, J., Cook, D., Lewin-Koh, N., Majure, J. J., and Megretskaia, I. (2000). Linking ArcView and XGobi: Insight behind the front end. Journal of Computational and Graphical Statistics, 9(3):470–490.

Tao, C., Kainz, W. & Zuidam, R.A.(1996). Cupoling GIS and Environmental Modelling: The Implications for Spatio-Temporal Data Modelling. Internationals Archieves of Photogrammetry and Remote Sensing. Vol. XXXI, Part B3 Vienna

Tarumi, T., Iizuka, M., Mori, Y., & Tanaka Y. (2003). Statistical software VASMM for variable selection in multivariate methods. In COMPSTAT 2002 Proceedings in Computational Statistics, pages 563–568. Physica, Heidelberg.

Tavakol, M., & Dennick, R. (2011). Making sense of Cronbach's alpha. Int. J. Medical Education, 2, 53-55. http://dx.doi.org/10.5116/ijme.4dfb.8dfd

Torun, A., & Duzgun, H. (2008). Relevance of Visual Exploration for Strengthening Spatial Thinking and Spatial Knowledge Exploration. In XXIst ISPRS Congress (p. China. Commission VIII papers). Beijing: China. Commission VIII papers.

Tutorialspoint.com,. (2015). Tutorials for Pay per Click, Accounting, Sqoop, XML, Software Engineering and many more latest technologies. Retrieved 11 February 2015, from http://tutorialspoint.com

Ungerer, M., & Goodchild, M. (2002). Integrating spatial data analysis and GIS: a new implementation using the Component Object Model (COM). International Journal Of Geographical Information Science, 16(1), 41-53. http://dx.doi.org/10.1080/13658810110095066

Urbanek, S. (2003). Rserve A Fast Way to Provide R Functionality to Applications. In 3rd International Workshop on Distributed Statistical (pp. 1609-395X). Augsburg. Retrieved from http://www.ci.tuwien.ac.at/Conferences/DSC-2003/

Venables, W. N., Smith, D. M., and the R Development Core Team, (2008). An Introduction to R, Notes on R: A Programming Environment for Data Analysis and Graphics Version 2.8.1.

Wikipedia, (2015). Software extension. [online] Available at: http://en.wikipedia.org/wiki/Software_extension [Accessed 12 Feb. 2015].

Wise, S., Haining, R., & Ma, J. (2001). Providing spatial statistical data analysis functionality for the GIS user: the SAGE project. International Journal Of Geographical Information Science, 15(3), 239-254. http://dx.doi.org/10.1080/13658810151072877

Zhang, Z., & Griffith, D. (1997). Developing user-friendly spatial statistical analysis modules for GIS: An example using ArcView. Computers, Environment And Urban Systems, 21(1), 5-29. http://dx.doi.org/10.1016/s0198-9715(97)00011-2

APPENDIX A

R SCRIPT OF EACH R GEOSPATIAL NON-GEOSPATIAL TECHNIQUES

Table A 1. Kernel Density Estimation R Script

Kernel Density Estimation RScripts

require(tensor)
require(abind)
require(nlme)
require(mgcv)
require(spatstat)
require(maptools)
require(rgdal)
require(foreign)
require(lattice)
require(sp)
require(raster)
packageExist<-require(Runiversal)
if(!packageExist){
install.packages("Runiversal", repos=" http://cran.r-project.org")
}
 	
dataOgr <- readOGR("C:/udigData/data" ,"Income")
dataOgr1<-spTransform(dataOgr,CRS("+proj=utm +zone=36 +ellps=WGS84 +datum=WGS84 +units=m +no_defs"))
data <-as.ppp(dataOgr)
w <- data$window
data <- ppp(x=data$x, y=data$y, window=w)
sigma1 <- (1/8) * min(abs(diff(w$xrange)), abs(diff(w$yrange)))
data2 <-as.ppp(dataOgr1)
w1 <- data2$window

Table A 1. (Continued)

```
data2 <- ppp(x=data2$x, y=data2$y, window=w1)
sigma2 <- (1/8) * min(abs(diff(w1$xrange)), abs(diff(w1$yrange)))
sigma3 <-(sigma1/sigma2)*4000
capture.output(sigma3, file="C:\\udigData\\output\\sigma320151131193426.txt")
dataOgr <- readOGR("C:/udigData/data" ,"Income")
proj4string(dataOgr)
data <-as.ppp(dataOgr)
box <- bbox(dataOgr)
w <- owin(xrange=c(box[1],box[3]), yrange=c(box[2],box[4]))
data <- ppp(x=data$x, y=data$y, window=w)
dataOgr1<-spTransform(dataOgr,CRS("+proj=utm +zone=36 +ellps=WGS84
+datum=WGS84 +units=m +no_defs"))
box1 <- bbox(dataOgr1)
ldx<-(box1[3]-box1[1])/2048
ldy<-(box1[4]-box1[2])/2048
a<-ceiling(ldx)
b<-ceiling(ldy)
spatstat.options(npixel=c(a,b))
analysis <- density.ppp(data , 0.04823039
, weights=dataOgr$incomeTLpe, edge=TRUE)
writeGDAL(as.SpatialGridDataFrame.im(analysis),"C:/udigData/output/Income20
151131193448.tiff","GTiff")
```

Table A 2. Adaptive Density Estimation RScripts

Adaptive Density Estimation RScripts

```
require(tensor)
require(abind)
require(nlme)
require(mgcv)
require(spatstat)
require(maptools)
require(rgdal)
require(foreign)
require(lattice)
require(sp)
require(raster)
```

Table A 2**.** (Continued)

```
packageExist<-require(Runiversal)
if(!packageExist){
install.packages("Runiversal", repos=" http://cran.r-project.org")
}

Xogr <- readOGR("C:/udigData/data" ,"Income")
X <-as.ppp(Xogr)
analysis <- adaptive.density(X , f=0.1)
writeGDAL(as.SpatialGridDataFrame.im(analysis),"C:/udigData/output/Income20
15113119385.tiff","GTiff")
```

Table A 3. Uniform Intensity RScripts

Uniform Intensity RScripts

```
require(tensor)
require(abind)
require(nlme)
require(mgcv)
require(spatstat)
require(maptools)
require(rgdal)
require(Runiversal)
require(foreign)
require(lattice)
require(sp)
require(raster)

Xogr <- readOGR("C:/udigData/data" ,"Income")
X <-as.ppp(Xogr)
analysis <- summary(X)
capture.output(analysis, file="C:\\udigData\\output\\uniform20151131193919.txt")
```

Table A 4. Kriging RScripts – Variogram

Kriging RScripts - Variogram
require(spatstat) require(maptools) require(rgdal) require(sp) data(meuse) class(meuse) names(meuse) coordinates(meuse) = ~x+y class(meuse) summary(meuse) lzn.vgm = variogram(log(zinc)~1, meuse) lzn.vgm = variogram(log(zinc)~1, meuse) lzn.fit = fit.variogram(lzn.vgm, model = vgm(1, "Sph", 900, 1)) lzn.fit lznr.vgm = variogram(log(zinc)~sqrt(dist), meuse) lznr.fit = fit.variogram(lznr.vgm, model = vgm(1, "Exp", 300, 1)) lznr.fit plot(lznr.vgm, lznr.fit)

Table A 5. Kriging RScripts – Interpoation

Kriging RScripts - Interpoation
require(spatstat) require(maptools) require(rgdal) require(sp) require(raster) data(meuse) class(meuse) names(meuse) coordinates(meuse) = ~x+y class(meuse) summary(meuse) lzn.kriged = krige(log(zinc)~1, meuse, meuse.grid, model = lzn.fit) spplot(lzn.kriged["var1.pred"])

Table A 6. Simulation Envelope for CSR RScripts

Simulation Envelope for CSR RScripts
require(tensor) require(abind) require(nlme) require(mgcv) require(spatstat) require(maptools) require(rgdal) require(foreign) require(lattice) require(sp) require(raster) Xogr <- readOGR("C:/udigData/data" ,"Income") X <-as.ppp(Xogr) UnmarkX <-unmark(X) analysis <- envelope(UnmarkX, fun=Gest, nsim=9) plot(analysis)

Table A 7. Fitted Poisson Model RScripts

Fitted Poisson Model RScripts
require(tensor) require(abind) require(nlme) require(mgcv) require(spatstat) require(maptools) require(rgdal) require(foreign) require(lattice) require(sp) require(raster) dataOgr <- readOGR("C:/udigData/data" ,"Income") proj4string(dataOgr) data <-as.ppp(dataOgr) box <- bbox(dataOgr)

Table A 7. (Continued)

```
w <- owin(xrange=c(box[1],box[3]), yrange=c(box[2],box[4]))
dataW <- ppp(x=data$x, y=data$y, window=w)
fit <- ppm(dataW, trend=~1, interaction=Poisson())
trend <- predict(fit, type="trend")
capture.output(fit, file="C:\\udigData\\output\\fit2015113119448.txt")
writeGDAL(as.SpatialGridDataFrame.im(trend),"C:/udigData/output/Income2015
1131194418.tiff","GTiff")
```

Table A 8. G Estimation RScripts

G Estimation RScripts

```
require(spatstat)
require(maptools)
require(rgdal)
require(sp)

Xogr <- readOGR("C:/udigData/data" ,"Income")
X <-as.ppp(Xogr)
analysis <- Gest(X , correction=c("han","km","rs"))
plot(analysis)
```

Table A 9. F Estimation RScripts

F Estimation RScripts

```
require(spatstat)
require(maptools)
require(rgdal)
require(sp)

Xogr <- readOGR("C:/udigData/data" ,"Income")
X <-as.ppp(Xogr)
analysis <- Fest(X , correction=c("cs","km","rs"))
plot(analysis)
```

Table A 10. F Estimation RScripts

L Estimation RScripts
require(spatstat) require(maptools) require(rgdal) require(sp) Xogr <- readOGR("C:/udigData/data" ,"Income") X <-as.ppp(Xogr) analysis <- Lest(X) plot(analysis)

Table A 11. K Estimation RScripts

K Estimation RScripts
require(spatstat) require(maptools) require(rgdal) require(sp) Xogr <- readOGR("C:/udigData/data" ,"Income") X <-as.ppp(Xogr) analysis <- Kest(X) plot(analysis)

Table A 12. Komolov Smirnov Test RScripts

Komolov Smirnov Test RScripts
require(tensor) require(abind)

Table A 12. (Continued)

```
require(spatstat)
require(maptools)
require(rgdal)
require(sp)

Xogr <- readOGR("C:/udigData/data" ,"Income")
X <-as.ppp(Xogr)
fun <-function(x,y){x}
analysis <- kstest(X, fun)
plot(analysis)
capture.output(analysis, file="C:\\udigData\\output\\kstest2015113119503.txt")
```

Table A 13. PointinPoly (Density Estimation in Buffered Areas) RScripts

PointinPoly (Density Estimation in Bufferede Areas) RScripts

```
require(rgeos)
require(MASS)
require(RColorBrewer)
require(sp)
require(maptools)
require(GISTools)
require(rgdal)

areaDbf <- read.dbf("C:/udigData/data/Buffer1Poly.dbf" , as.is = TRUE)
areaOgr <- readOGR("C:/udigData/data" ,"Buffer1Poly")
pointOgr <- readOGR("C:/udigData/data" ,"Income")
getClass("Polygon")
sapply(slot(areaOgr, "polygons"), function(x) length(slot(x, "Polygons")))
sapply(slot(areaOgr, "polygons"), function(x) {
xi <- slot(x, "Polygons")
any(sapply(xi, slot, "hole"))
})
ur.area<-sapply(slot(areaOgr, "polygons"), slot, "area")
str(ur.area)
a<- ur.area
areaDbf$AreaofPolygon <- a
write.dbf(areaDbf, "C:/udigData/data/Buffer1Poly.dbf")
count <- poly.counts(pointOgr, areaOgr)
areaDbf$NumberofPoint <- count
```

Table A 13. (Continued)

```
write.dbf(areaDbf, "C:/udigData/data/Buffer1Poly.dbf")
nokbolalan <- count/a
areaDbf$Density <- nokbolalan
write.dbf(areaDbf, "C:/udigData/data/Buffer1Poly.dbf")
```

Table A 14. Geographical Weghted Regression Model (GWR) RScripts

Geographical Weghted Regression Model (GWR) RScripts

```
require(rgeos)
require(MASS)
require(RColorBrewer)
require(sp)
require(maptools)
require(GISTools)
require(rgdal)
require(foreign)

areaDbf <- read.dbf("C:/udigData/data/Buffer1Poly.dbf" , as.is = TRUE)
areaOgr <- readOGR("C:/udigData/data" ,"Buffer1Poly")
pointOgr <- readOGR("C:/udigData/data" ,"Income")
getClass("Polygon")
sapply(slot(areaOgr, "polygons"), function(x) length(slot(x, "Polygons")))
sapply(slot(areaOgr, "polygons"), function(x) {
xi <- slot(x, "Polygons")
any(sapply(xi, slot, "hole"))
})
ur.area<-sapply(slot(areaOgr, "polygons"), slot, "area")
str(ur.area)
a<- ur.area
areaDbf$AreaofPolygon <- a
write.dbf(areaDbf, "C:/udigData/data/Buffer1Poly.dbf")
count <- poly.counts(pointOgr, areaOgr)
areaDbf$NumberofPoint <- count
write.dbf(areaDbf, "C:/udigData/data/Buffer1Poly.dbf")
nokbolalan <- count/a
areaDbf$Density <- nokbolalan
write.dbf(areaDbf, "C:/udigData/data/Buffer1Poly.dbf")
```

Table A 15. Spatial Regression Model (SRM) RScripts

Spatial Regression Model (SRM) RScripts
require(rgeos) require(MASS) require(RColorBrewer) require(sp) require(maptools) require(GISTools) require(rgdal) require(foreign) areaDbf <- read.dbf("C:/udigData/data/Buffer1Poly.dbf" , as.is = TRUE) areaOgr <- readOGR("C:/udigData/data" ,"Buffer1Poly") pointOgr <- readOGR("C:/udigData/data" ,"Income") getClass("Polygon") sapply(slot(areaOgr, "polygons"), function(x) length(slot(x, "Polygons"))) sapply(slot(areaOgr, "polygons"), function(x) { xi <- slot(x, "Polygons") any(sapply(xi, slot, "hole")) }) ur.area<-sapply(slot(areaOgr, "polygons"), slot, "area") str(ur.area) a<- ur.area areaDbf$AreaofPolygon <- a write.dbf(areaDbf, "C:/udigData/data/Buffer1Poly.dbf") count <- poly.counts(pointOgr, areaOgr) areaDbf$NumberofPoint <- count write.dbf(areaDbf, "C:/udigData/data/Buffer1Poly.dbf") nokbolalan <- count/a areaDbf$Density <- nokbolalan write.dbf(areaDbf, "C:/udigData/data/Buffer1Poly.dbf")

Table A 16. Histogram Graph RScripts

Histogram Graph RScripts
require(abind) require(deldir) require(nlme) require(mgcv)

Table A 16. (Continued)

```
require(lattice)
require(spatstat)
require(sp)
require(foreign)

dataOgr <- readOGR("C:/udigData/data" ,"Income")
t <- table(dataOgr$incomeTLpe)
pct <- round(t/sum(t)*100)
lbls <- paste("
" , t, "#" , "
" , pct , "%" , sep ="")
analysis <- barplot(t, col=rainbow(nrow(t)),names.arg=c(lbls), legend =
rownames(t), main="Histogram Graph")
dev.off()
q("yes")
```

Table A 17. Column Graph RScripts

Column Graph RScripts

```
require(abind)
require(deldir)
require(nlme)
require(mgcv)
require(lattice)
require(spatstat)
require(sp)
require(foreign)

dataOgr <- readOGR("C:/udigData/data" ,"Income")
analysis <- hist(dataOgr$incomeTLpe, breaks=8, col=2, labels=TRUE, xlab="X
Axis of Graph", ylab="Y Axis of Graph", main="Column Graph of Data",
border=3)
```

Table A 18. Pie Chart RScripts

Pie Chart RScripts
require(lattice) require(spatstat) require(sp) require(foreign) dataOgr <- readOGR("C:/udigData/data" ,"Income") t <- table(dataOgr$incomeTLpe) pct <- round(t/sum(t)*100) lbls <- paste(names(t) , " " , t, "#" , " " , pct , "%" , sep ="") analysis <- pie(t, labels=lbls,, col=rainbow(nrow(t)), main="Pie Chart of Data")

APPENDIX B

sda4uDig QUESTIONNAIRE FOR ONLINE USERS

sda4udig

sda4udig

sda4uDig Survey for End User

Dear User,

Before downloading the software, we ask to fill this survey which will be used for assessing the user requirements to be analyzed in the the Ph.D. thesis of Mahmut Cavur (mahmutcavur@gmail.com) entitled "Integration of Geospatial Techniques (R) into an Open-Source GIS Software (uDig)" which is supervised by Prof. Dr. H. Sebnem Duzgun (duzgun@metu.edu.tr) at Middle East Technical University, Geodetic and Geographic Information Technologies Program. Your feedback and insight will help developer to embrace a better solution for geospatial data analysis.

1. Contact

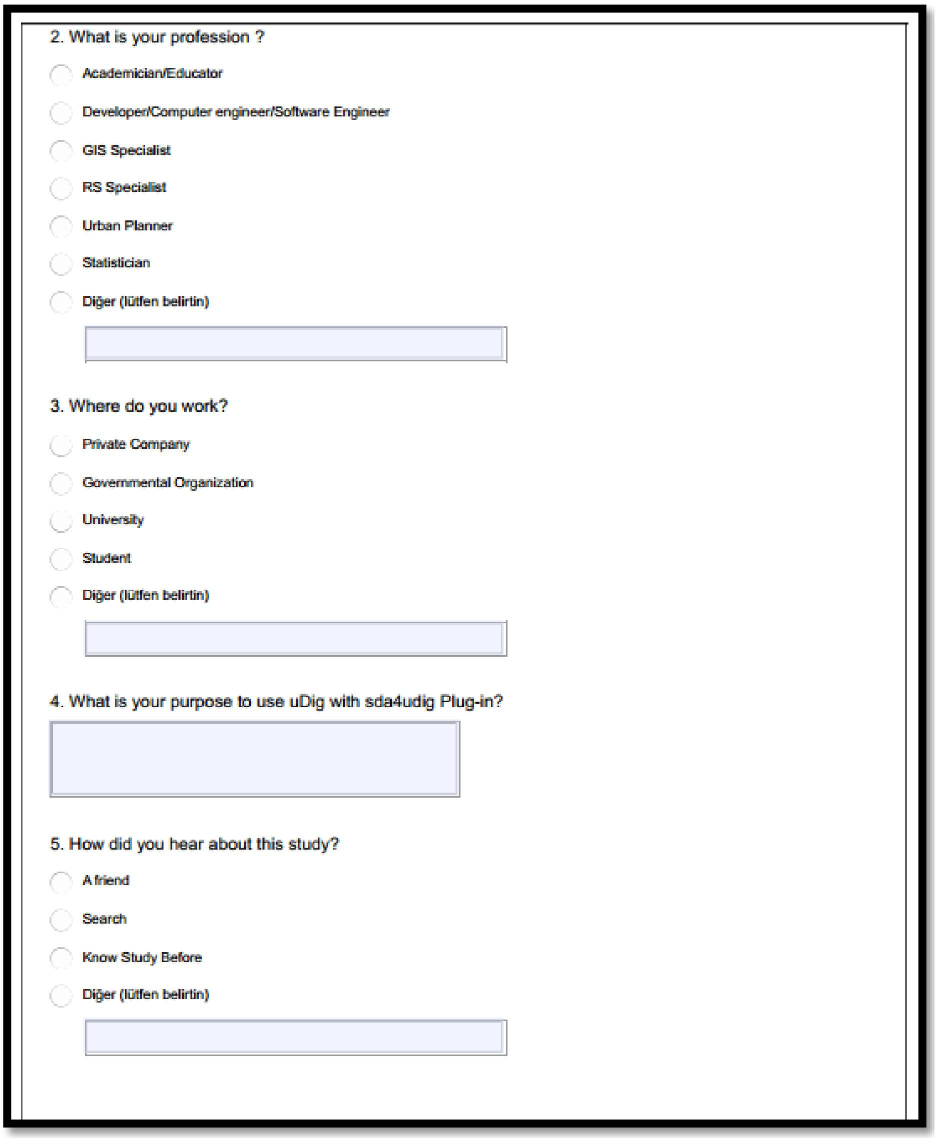
Name/Surname:

E-mail*:

Country:

Institue:

Figure B 1. sda4uDig Survey for Web-Site Visitors



2. What is your profession ?

- Academician/Educator
- Developer/Computer engineer/Software Engineer
- GIS Specialist
- RS Specialist
- Urban Planner
- Statistician
- Diğer (lütfen belirtin)

3. Where do you work?

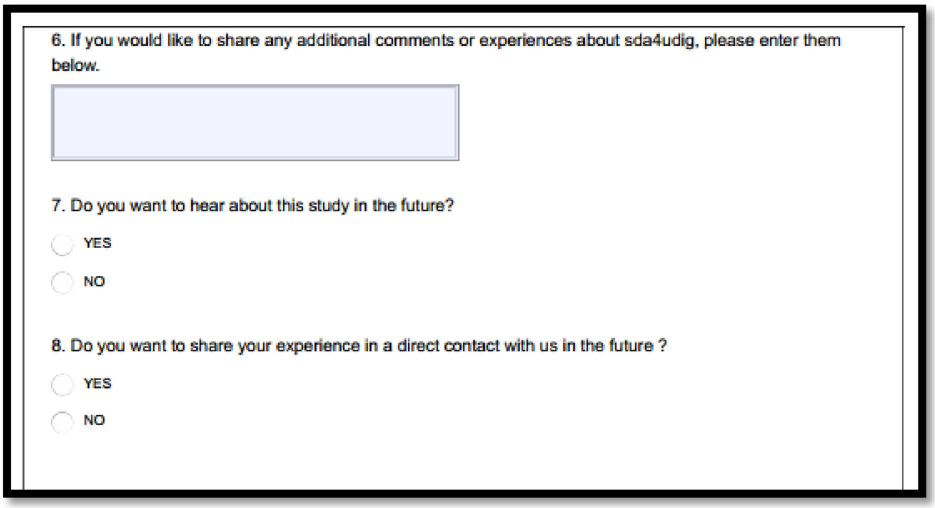
- Private Company
- Governmental Organization
- University
- Student
- Diğer (lütfen belirtin)

4. What is your purpose to use uDig with sda4udig Plug-in?

5. How did you hear about this study?

- A friend
- Search
- Know Study Before
- Diğer (lütfen belirtin)

Figure B 1. (Continued)


6. If you would like to share any additional comments or experiences about sda4udig, please enter them below.

7. Do you want to hear about this study in the future?

YES

NO

8. Do you want to share your experience in a direct contact with us in the future ?

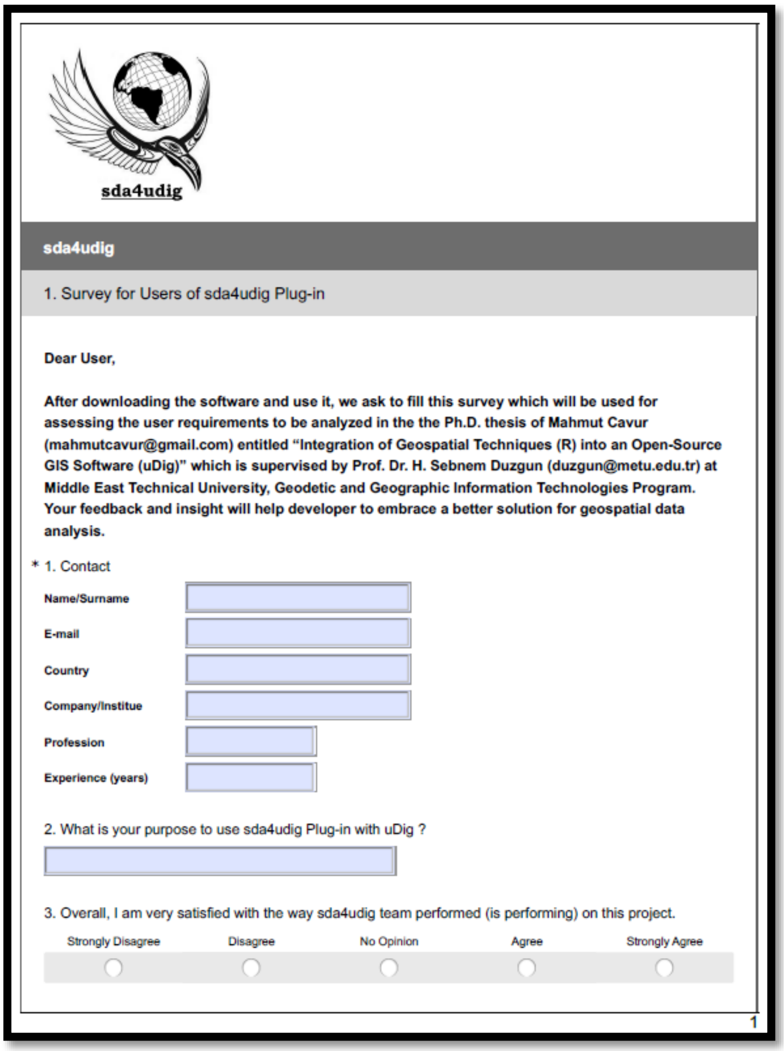
YES

NO

Figure B 1. (Continued)

APPENDIX C

sda4uDig QUESTIONNAIRE FOR HANDS-ON USERS



sda4udig

sda4udig

1. Survey for Users of sda4udig Plug-in

Dear User,

After downloading the software and use it, we ask to fill this survey which will be used for assessing the user requirements to be analyzed in the the Ph.D. thesis of Mahmut Cavur (mahmutcavur@gmail.com) entitled "Integration of Geospatial Techniques (R) into an Open-Source GIS Software (uDig)" which is supervised by Prof. Dr. H. Sebnem Duzgun (duzgun@metu.edu.tr) at Middle East Technical University, Geodetic and Geographic Information Technologies Program. Your feedback and insight will help developer to embrace a better solution for geospatial data analysis.

* 1. Contact

Name/Surname

E-mail

Country

Company/Institue

Profession

Experience (years)

2. What is your purpose to use sda4udig Plug-in with uDig ?

3. Overall, I am very satisfied with the way sda4udig team performed (is performing) on this project.

Strongly Disagree	Disagree	No Opinion	Agree	Strongly Agree
○	○	○	○	○

1

Figure C 1. Survey for Users of sda4uDig after Usage

4. How often do you typically use the Geospatial Data Analysis Software/Technique(s)?

Daily	Weekly	Once a month	Once a year	Do not use
○	○	○	○	○

5. Based on your awareness of sda4udig, is it better, the same, or worse than other brands of Spatial Data Analysis Software ?

Much Worse	Worse	About the same	Better	Much Better
○	○	○	○	○

6. Based on your experience with sda4udig, would you recommend this product to a friend?

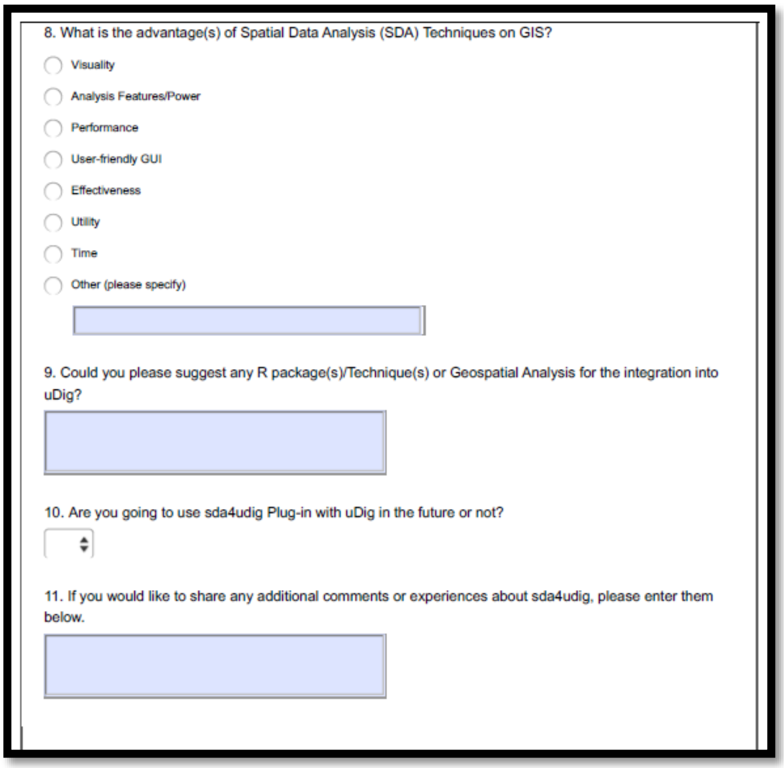
Definitely will not	Probably will not	Might or might not	Probably will	Definitely will
○	○	○	○	○

7. How satisfied are you…

	Strongly Disagree	Disagree	No Opinion	Agree	Strongly Agree
…with the appropriateness of the documentation to your needs?	○	○	○	○	○
…with the quality of the sda4udig delivered with your requirements?	○	○	○	○	○
…with the usability of the software provided?	○	○	○	○	○
…with the performance of the udig with sda4udig software?	○	○	○	○	○
…with the easiness of the sda4udig plug-in?	○	○	○	○	○
… with the number of Spatial Data Analysis techniques?	○	○	○	○	○
…with the Spatial Data Analysis techniques on GIS?	○	○	○	○	○

2

Figure C 1. (Continued)



8. What is the advantage(s) of Spatial Data Analysis (SDA) Techniques on GIS?

- Visuality
- Analysis Features/Power
- Performance
- User-friendly GUI
- Effectiveness
- Utility
- Time
- Other (please specify)

9. Could you please suggest any R package(s)/Technique(s) or Geospatial Analysis for the integration into uDig?

10. Are you going to use sda4udig Plug-in with uDig in the future or not?

11. If you would like to share any additional comments or experiences about sda4udig, please enter them below.

Figure C 1. (Continued)

APPENDIX D

sda4uDig USER TASK

Dear user,

This user task is a part of Phd study carried out by Mahmut Çavur and his advisor Prof. Dr. H.Şebnem Düzgün in Department of Geodetic and Geographic Information Technology at METU.

In the content of this task study, there are three case studies and their steps are shown in the following figures. As a user, you need to follow each step and apply them to get the same result. After completing three case studies, you need to fill the questionnaire survey which is provided for you. The questionnaire needs to be filled with respect to experience with this software and your past experince about SDA or GIS.

Thank you very much because of attendance and valuable feedback…

Kernel Density Estimation (Çekirdek Yoğunluğu Tahmin Modeli) Method:

<u>First Step:</u> Add Data

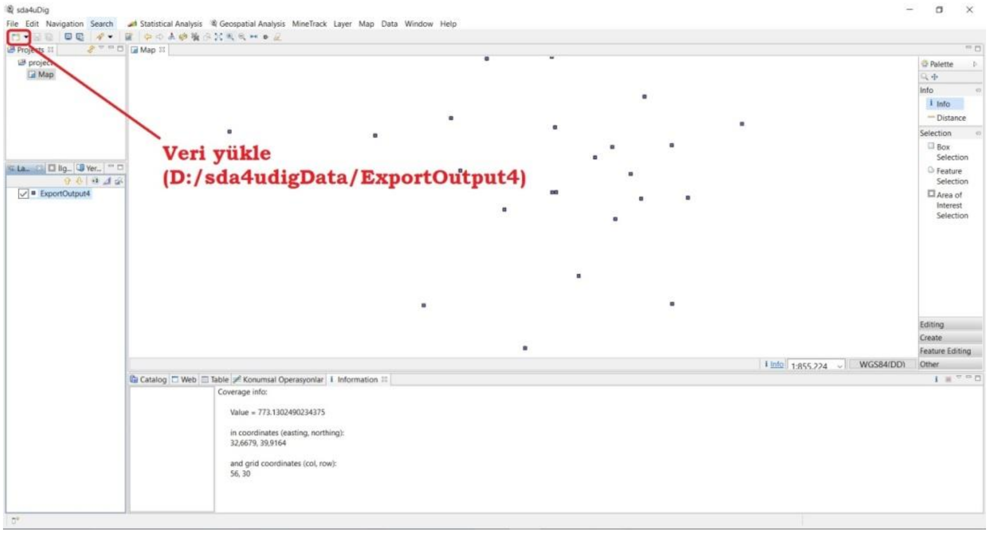


Figure D 1. Adding Data into uDig MapView

<u>Second Step: </u>KDE Technique Application

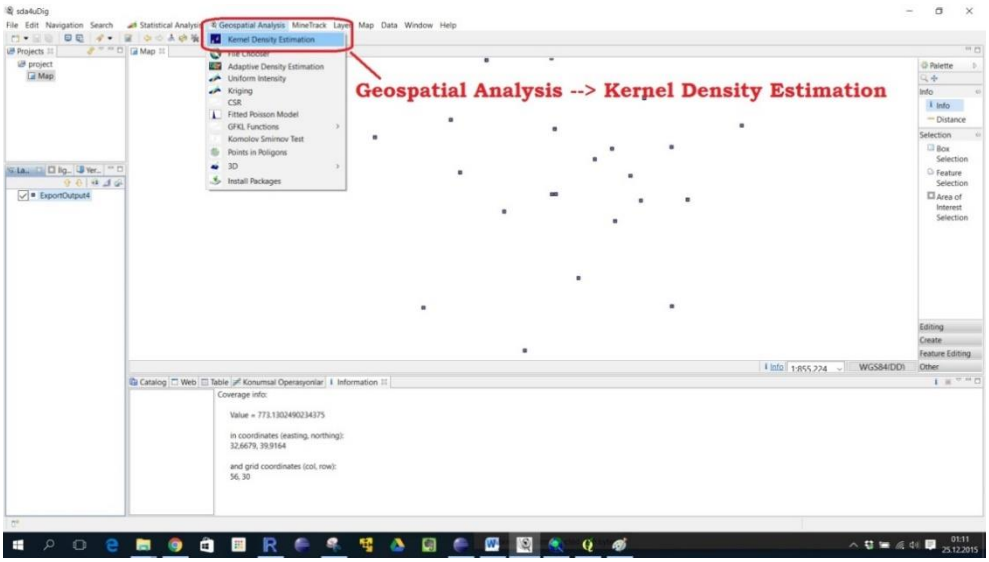


Figure D 2. KDE Tehcnique on Geospatial Menu on uDig

Third Step: Parameter Definition

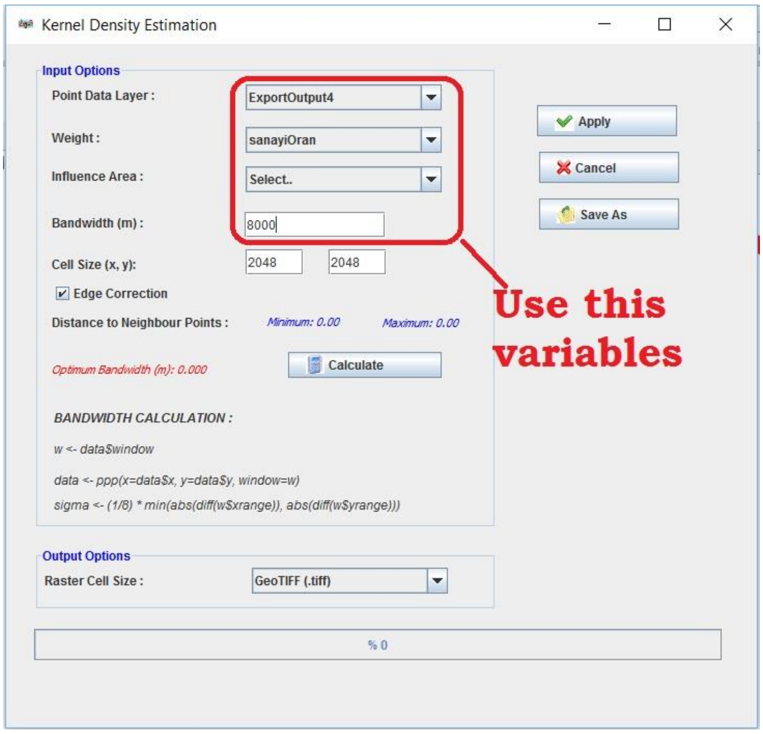


Figure D 3. KDE Input GUI

Fourth Step: Black-White Result

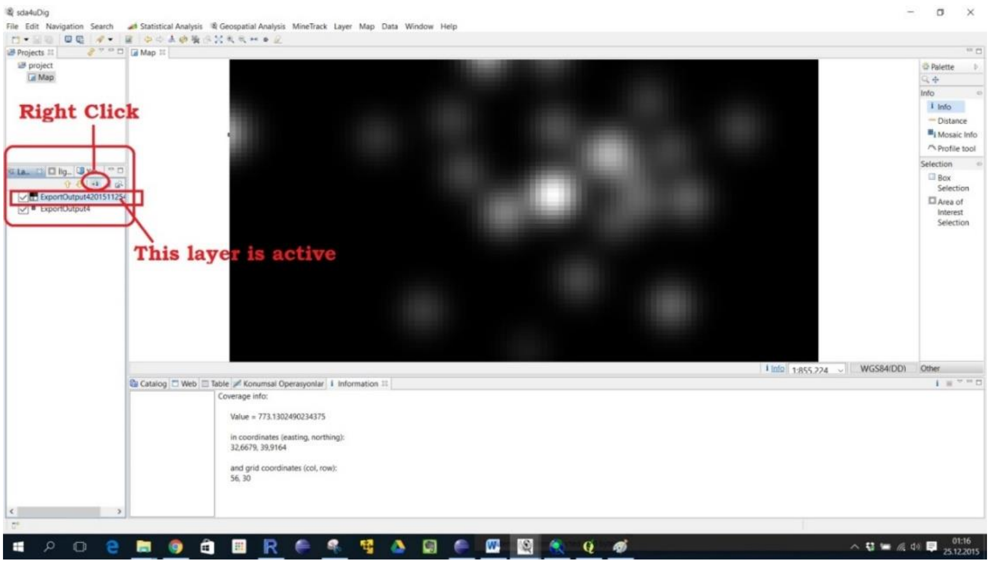


Figure D 4. KDE Result – Black and White

Fifth Step: Result Classification

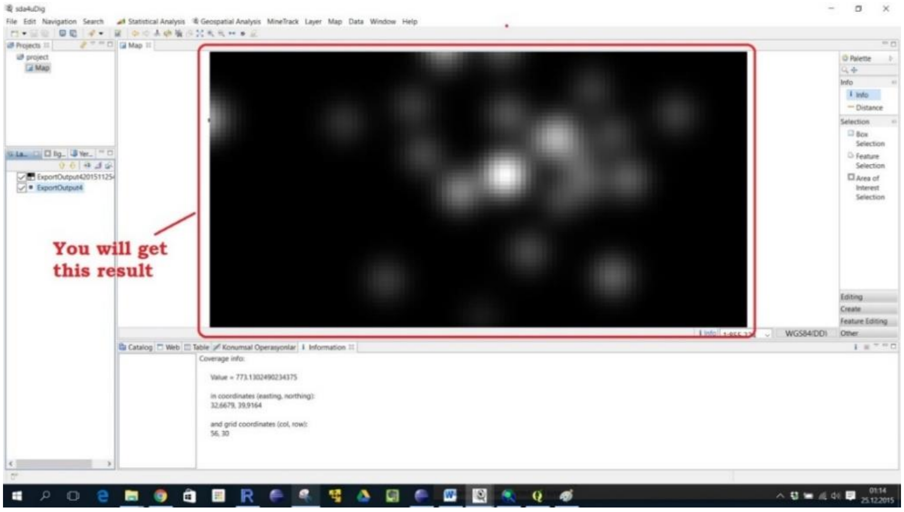


Figure D 5. Classification of Result

Sixth Step: Result Classification - Logarithmic

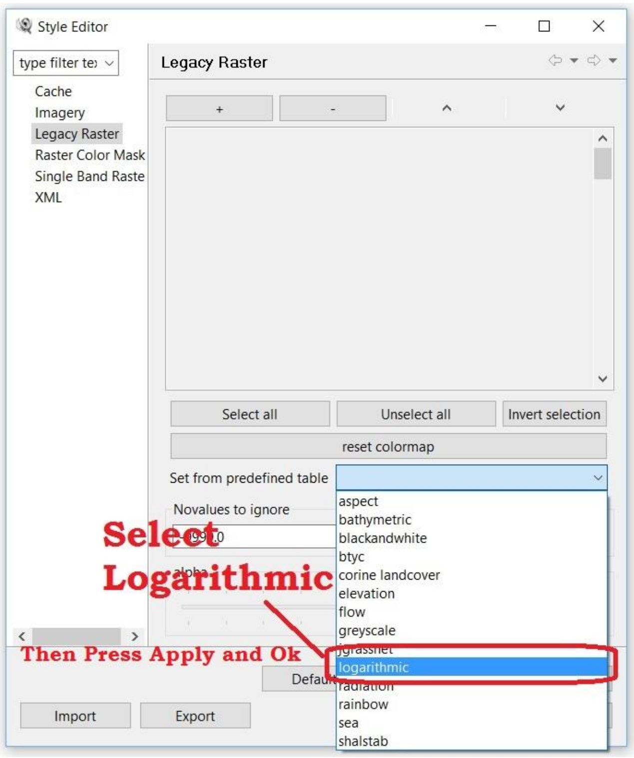


Figure D 6. Classification Wizard

Seventh Step: Final Result of KDE

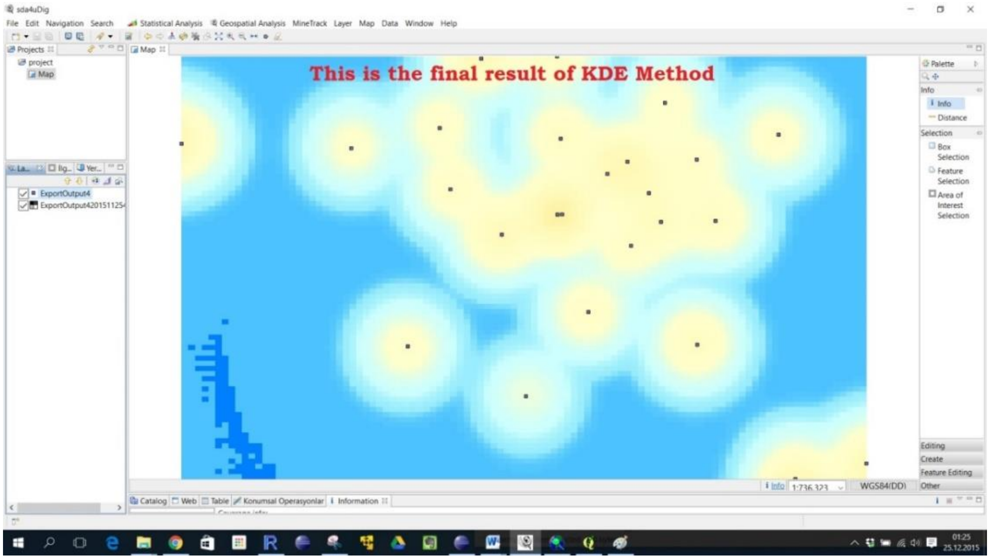


Figure D 7. Final Result of KDE

K Function

First Step: Add Data

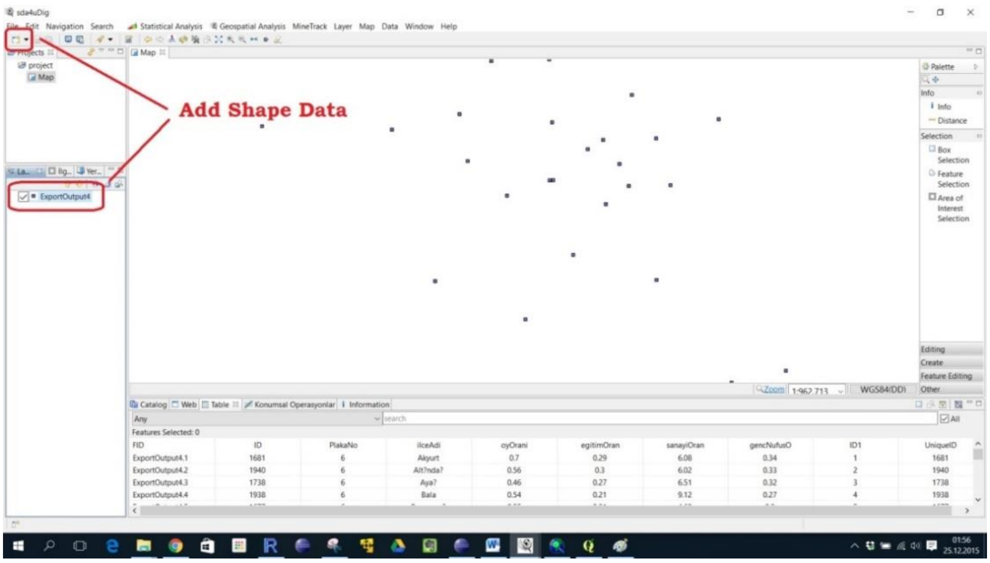


Figure D 8. Adding Data into uDig MapView

<u>Second Step:</u> Parameter Definition

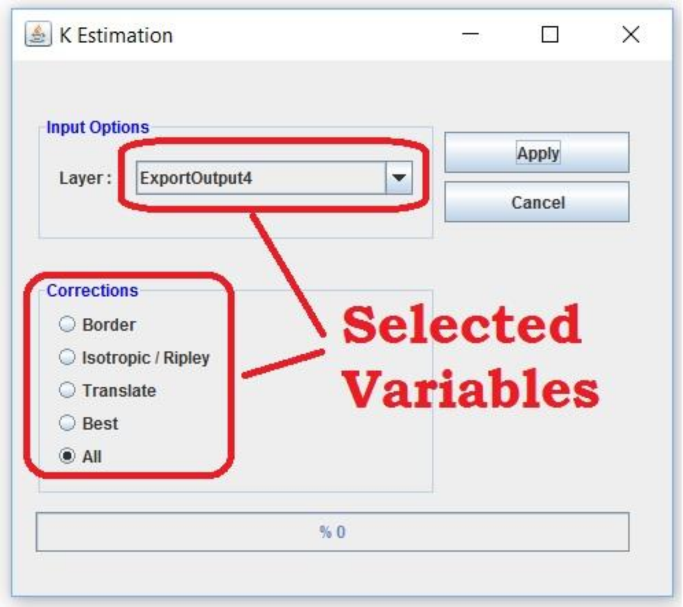


Figure D 9. K Estimaiton Input GUI

<u>Third Step:</u> Analysis Result

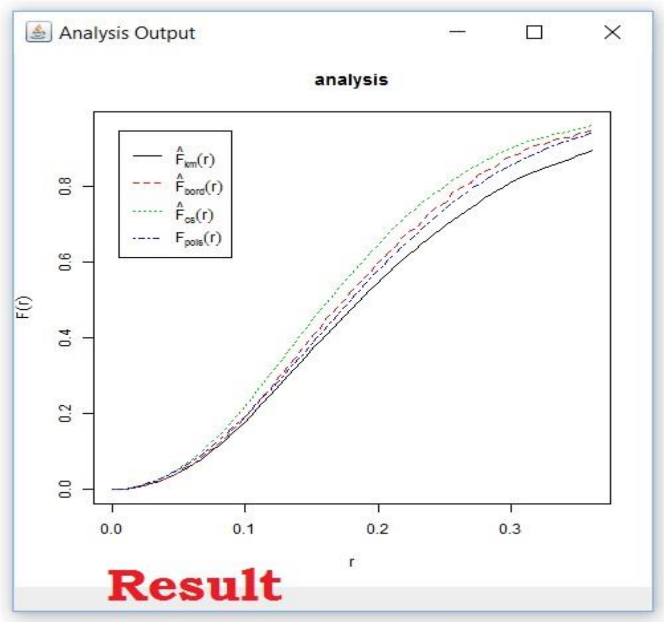


Figure D 10. K Estimation Analysis Result

Column Chart

First Step: Add Data

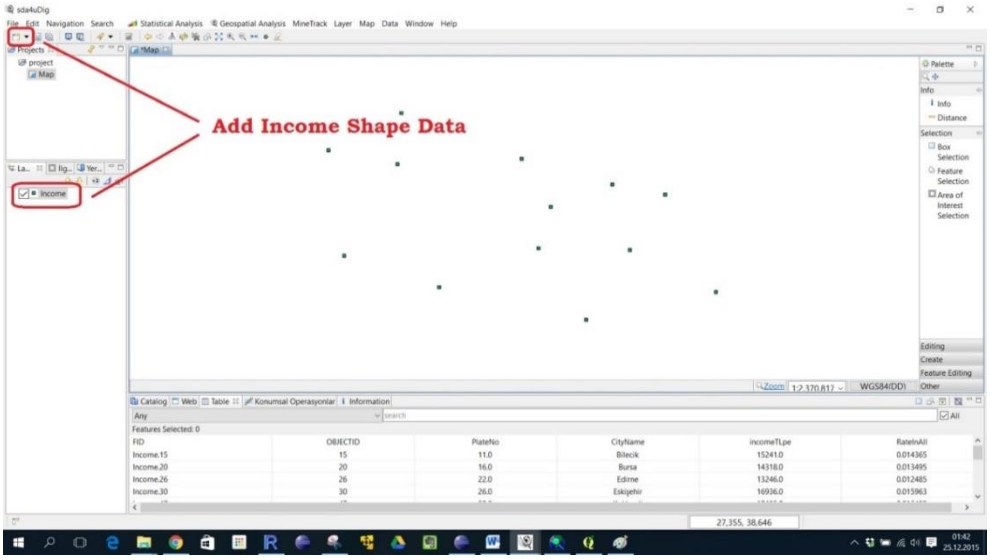


Figure D 11. Adding Data into uDig MapView

Second Step: Column Chart Technique Application

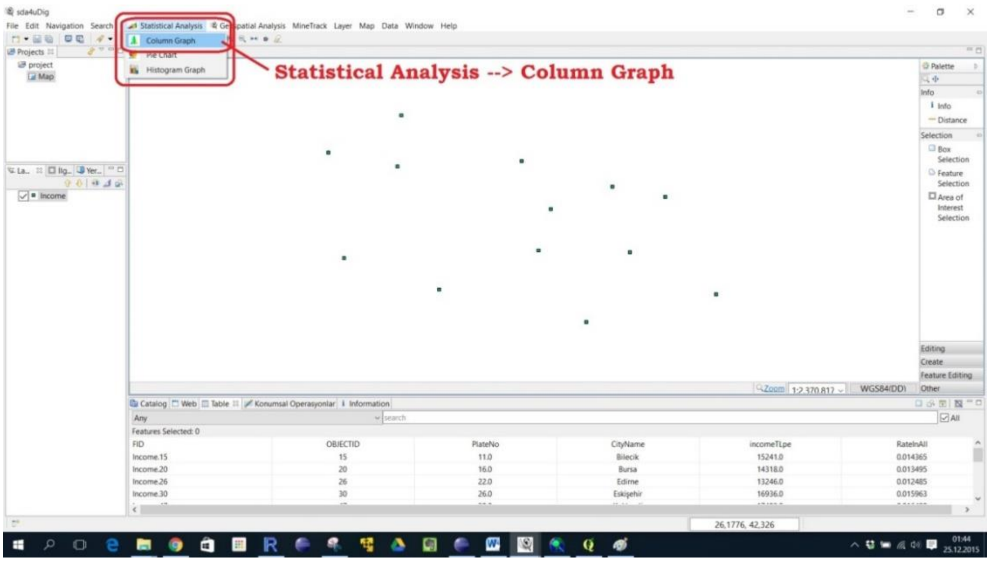


Figure D 12. Column Chart in Statistical Menu on uDig

<u>Third Step:</u> Variable Definition

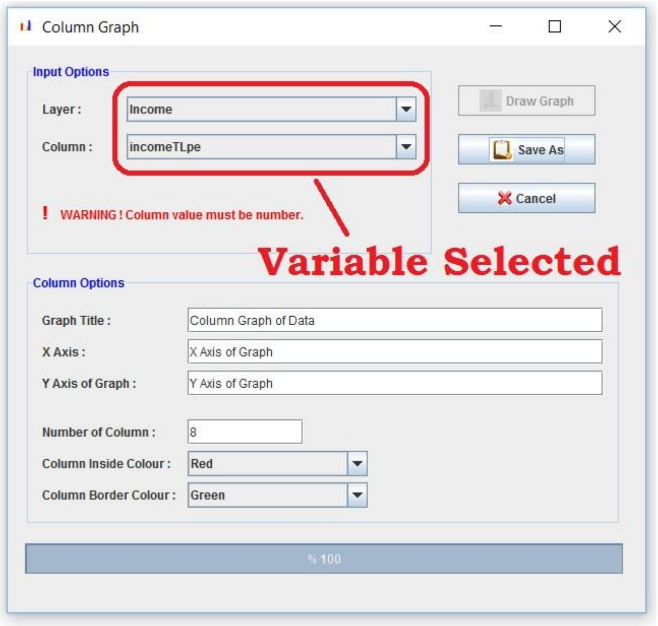


Figure D 13. Column Chart Input GUI

<u>Fourth Step:</u> Analysis Result

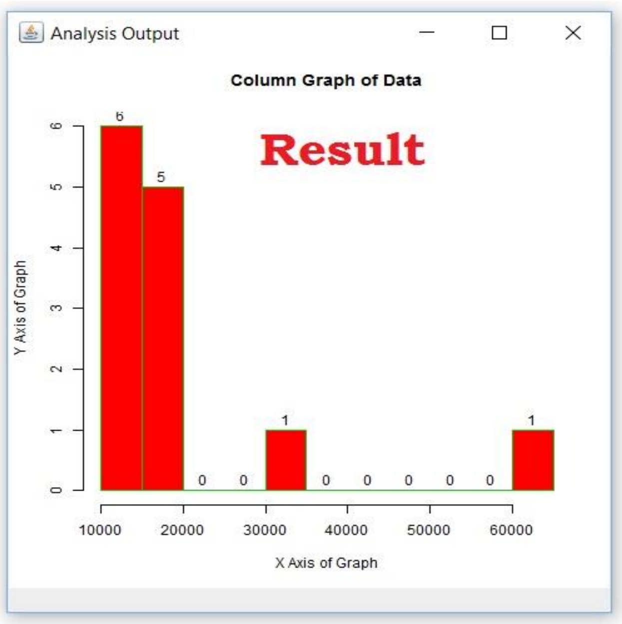


Figure D 14. Column Chart Analysis Result

APPENDIX E

SSPS RESULT

HYPOTHESIS 1

Table E 1. Descriptive Statistics of Hypothesis 1

	Mean	Std. Deviation	N
Satisfaction_3	4,1875	,69270	32
Usage_Frequency_4	3,5313	1,21773	32

Table E 2. Correlations of Hypothesis 1

		Satisfaction_3	Usage_Frequency_4
Pearson Correlation	Satisfaction_3	1,000	-,084
	Usage_Frequency_4	-,084	1,000
Sig. (1-tailed)	Satisfaction_3	.	,324
	Usage_Frequency_4	,324	.
N	Satisfaction_3	32	32
	Usage_Frequency_4	32	32

Table E 3. Variables Entered/Removed (b) of Hypothesis 1

Model	Variables Entered	Variables Removed	Method
1	Usage_Frequency_4(a)	.	Enter

a All requested variables entered.

b Dependent Variable: Satisfaction_3

Table E 4. Model Summary (b) of Hypothesis 1

Model	R	R Square	Adjusted R Square	Std. Error of the Estimate
1	,084(a)	,007	-,026	,70169

a Predictors: (Constant), Usage_Frequency_4

b Dependent Variable: Satisfaction_3

Table E 5. ANOVA (b)

Model		Sum of Squares	df	Mean Square	F	Sig.
1	Regression	,104	1	,104	,211	,649(a)
	Residual	14,771	30	,492		
	Total	14,875	31			

a Predictors: (Constant), Usage_Frequency_4

b Dependent Variable: Satisfaction_3

Table E 6. Coefficients (a) of Hypothesis 1

Model		Unstandardized Coefficients		Standard. Coeff.	t	Sig.	95% Confidence Interval for B	
		B	Std. Error	Beta	Lower Bound	Upper Bound	B	Std. Error
1	(Constant)	4,356	,386		11,286	,000	3,567	5,144
	Usage_Frequency_4	-,048	,103	-,084	-,460	,649	-,259	,164

a Dependent Variable: Satisfaction_3

Table E 7. Residuals Statistics (a) of Hypothesis 1

	Minimum	Maximum	Mean	Std. Deviation	N
Predicted Value	4,1176	4,3555	4,1875	,05795	32
Residual	-2,11761	,88239	,00000	,69028	32
Std. Predicted Value	-1,206	2,900	,000	1,000	32
Std. Residual	-3,018	1,258	,000	,984	32

a Dependent Variable: Satisfaction_3

CHARTS

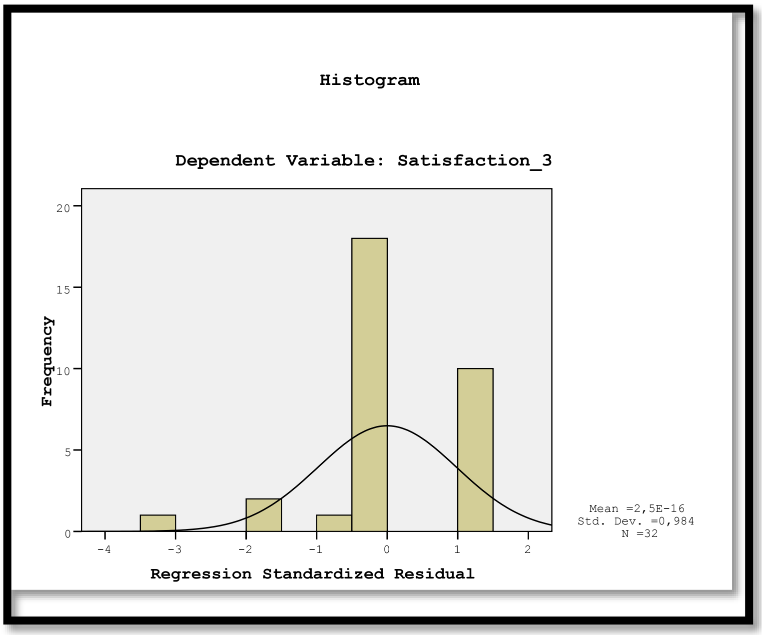


Figure E 1. Histogram Graph of Hypothesis 1

HYPOTHESIS 2

Table E 8. Descriptives of Hypothesis 2 that shows Satisfaction Level

	N	Mean	Std. Deviation	Std. Error	95% Confidence Interval for Mean		Min	Max
	Lower Bound	Upper Bound	Lower Bound	Upper Bound	Lower Bound	Upper Bound	Lower Bound	Upper Bound
No Opinion	4	3,7500	,50000	,25000	2,9544	4,5456	3,00	4,00
Agree	22	4,2273	,52841	,11266	3,9930	4,4616	3,00	5,00
Strongly Agree	5	4,8000	,44721	,20000	4,2447	5,3553	4,00	5,00
Total	31	4,2581	,57548	,10336	4,0470	4,4692	3,00	5,00

Table E 9. Test of Homogeneity of Variances of Hypothesis 2

Levene Statistic	df1	df2	Sig.
,257	2	28	,775

Table E 10. ANOVA of Hypothesis 2

	Sum of Squares	df	Mean Square	F	Sig.
Between Groups	2,522	2	1,261	4,762	,017
Within Groups	7,414	28	,265		
Total	9,935	30			

Table E 11. Robust Tests of Equality of Means of Hypothesis 2

	Statistic(a)	df1	df2	Sig.
Welch	5,200	2	6,467	,045
Brown-Forsythe	5,405	2	9,402	,027

a Asymptotically F distributed.

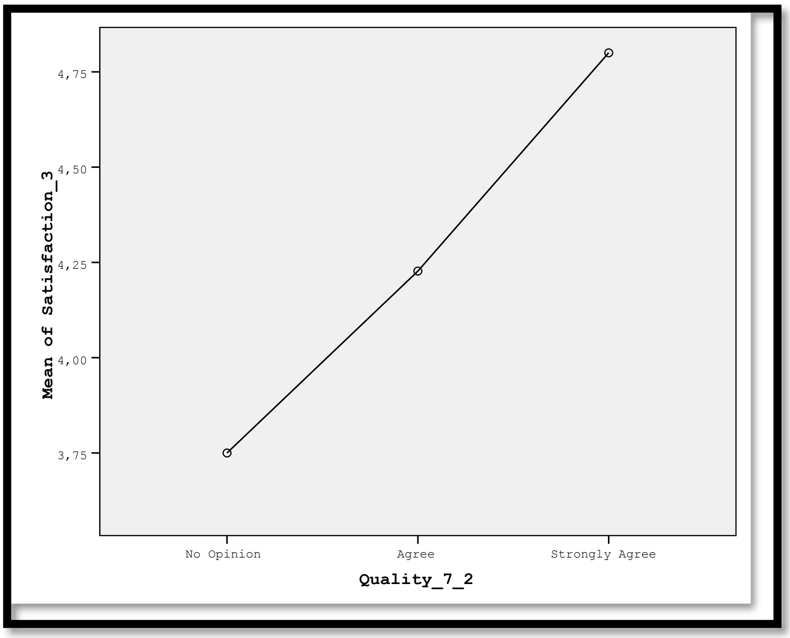


Figure E 2. Means Plots of Hypothesis 2

HYPOTHESIS 3

Table E 12. Descriptives of Hypothesis 3 that shows Satisfaction Level

	N	Mean	Std. Deviation	Std. Error	95% Confidence Interval for Mean		Min	Max
	Lower Bound	Upper Bound	Lower Bound	Upper Bound	Lower Bound	Upper Bound	Lower Bound	Upper Bound
Disagree	1	3,0000	.	.	.	.	3,00	3,00
No Opinion	7	4,2857	,75593	,28571	3,5866	4,9848	3,00	5,00
Agree	19	4,2632	,45241	,10379	4,0451	4,4812	4,00	5,00
Strongly Agree	4	4,5000	,57735	,28868	3,5813	5,4187	4,00	5,00
Total	31	4,2581	,57548	,10336	4,0470	4,4692	3,00	5,00

Table E 13. Test of Homogeneity of Variances of Hypothesis 3

Levene Statistic	df1	df2	Sig.
2,209(a)	2	27	,129

a Groups with only one case are ignored in computing the test of homogeneity of variance for Satisfaction_3.

Table E 14. ANOVA of Hypothesis 3

	Sum of Squares	df	Mean Square	F	Sig.
Between Groups	1,823	3	,608	2,022	,134
Within Groups	8,113	27	,300		
Total	9,935	30			

Table E 15. Robust Tests of Equality of Means (b) of Hypothesis 3

	Statistic(a)	df1	df2	Sig.
Welch	.	.	.	.
Brown-Forsythe	.	.	.	.

a Asymptotically F distributed.

b Robust tests of equality of means cannot be performed for Satisfaction_3 because at least one group has the sum of case weights less than or equal to 1.

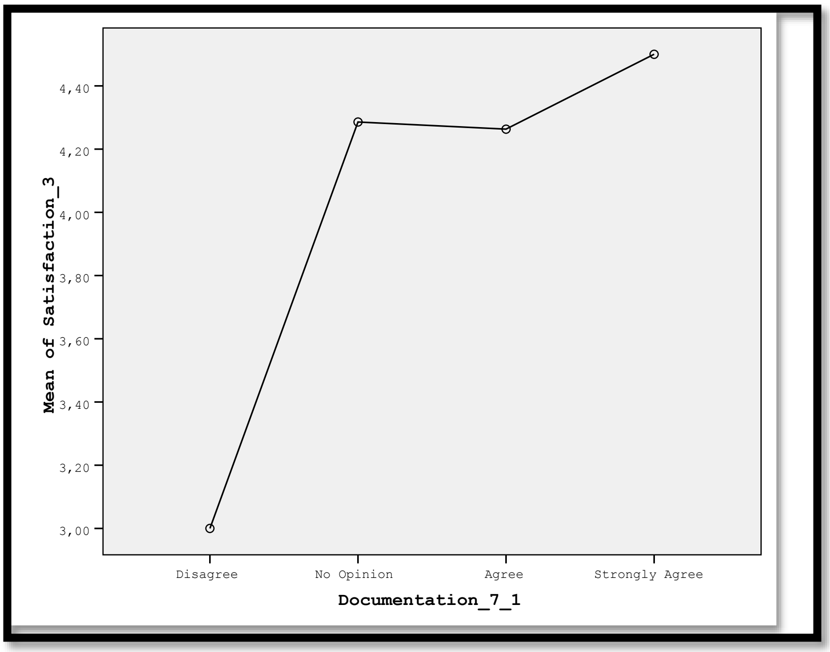


Figure E 3. Means Plots

HYPOTHESIS 4

Table E 16. Descriptives of Hypothesis 4 that shows Satisfaction Level

	N	Mean	Std. Deviation	Std. Error	95% Confidence Interval for Mean		Min	Max
	Lower Bound	Upper Bound	Lower Bound	Upper Bound	Lower Bound	Upper Bound	Lower Bound	Upper Bound
No Opinion	2	4,0000	,00000	,00000	4,0000	4,0000	4,00	4,00
Agree	21	4,1429	,57321	,12509	3,8819	4,4038	3,00	5,00
Strongly Agree	8	4,6250	,51755	,18298	4,1923	5,0577	4,00	5,00
Total	31	4,2581	,57548	,10336	4,0470	4,4692	3,00	5,00

Table E 17. Test of Homogeneity of Variances of Hypothesis 4

Levene Statistic	df1	df2	Sig.
1,576	2	28	,225

Table E 18. ANOVA of Hypothesis 4

	Sum of Squares	df	Mean Square	F	Sig.
Between Groups	1,489	2	,745	2,468	,103
Within Groups	8,446	28	,302		
Total	9,935	30			

Table E 19. Robust Tests of Equality of Means (b) of Hypothesis 4

	Statistic(a)	df1	df2	Sig.
Welch	.	.	.	.
Brown-Forsythe	.	.	.	.

a Asymptotically F distributed.

b Robust tests of equality of means cannot be performed for Satisfaction_3 because at least one group has 0 variance.

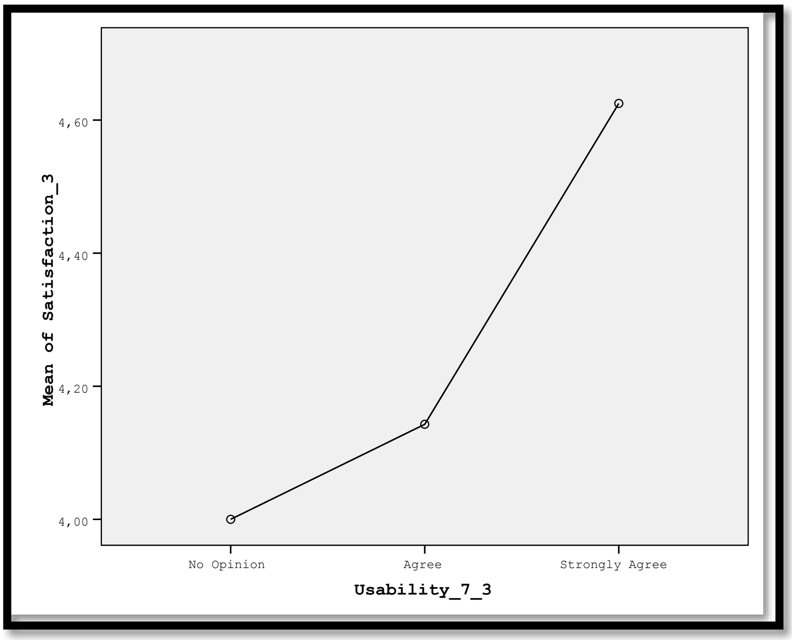


Figure E 4. Means Plots of Hypothesis 4

HYPOTHESIS 5

Table E 20. Descriptives of Hypothesis 5 that shows Satisfaction Level

	N	Mean	Std. Deviation	Std. Error	95% Confidence Interval for Mean		Min	Max
	Lower Bound	Upper Bound	Lower Bound	Upper Bound	Lower Bound	Upper Bound	Lower Bound	Upper Bound
Disagree	1	2,0000	.	.	.	.	2,00	2,00
No Opinion	14	4,0714	,61573	,16456	3,7159	4,4269	3,00	5,00
Agree	13	4,3077	,48038	,13323	4,0174	4,5980	4,00	5,00
Strongly Agree	4	4,7500	,50000	,25000	3,9544	5,5456	4,00	5,00
Total	32	4,1875	,69270	,12245	3,9378	4,4372	2,00	5,00

Table E 21. Test of Homogeneity of Variances of Hypothesis 5

Levene Statistic	df1	df2	Sig.
,042(a)	2	28	,959

a Groups with only one case are ignored in computing the test of homogeneity of variance for Satisfaction_3.

Table E 22. ANOVA of Hypothesis 5

	Sum of Squares	df	Mean Square	F	Sig.
Between Groups	6,427	3	2,142	7,101	,001
Within Groups	8,448	28	,302		
Total	14,875	31			

Table E 23. Robust Tests of Equality of Means(b) of Hypothesis 5

	Statistic(a)	df1	df2	Sig.
Welch	.	.	.	.
Brown-Forsythe	.	.	.	.

a Asymptotically F distributed.

b Robust tests of equality of means cannot be performed for Satisfaction_3 because at least one group has the sum of case weights less than or equal to 1.

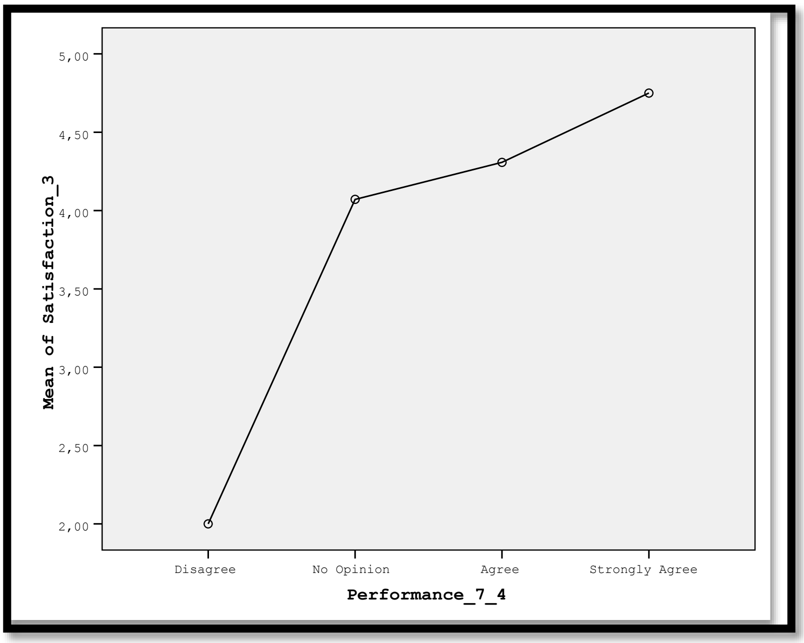


Figure E 5. Means Plots of Hypothesis 5

HYPOTHESIS 6

Table E 24. Descriptives of Hypothesis 6 that shows Satisfaction Level

	N	Mean	Std. Deviation	Std. Error	95% Confidence Interval for Mean		Min	Max
	Lower Bound	Upper Bound	Lower Bound	Upper Bound	Lower Bound	Upper Bound	Lower Boun d	Upper Bound
No Opinion	2	4,0000	,00000	,00000	4,0000	4,0000	4,00	4,00
Agree	16	3,8750	,71880	,17970	3,4920	4,2580	2,00	5,00
Strongl y Agree	14	4,5714	,51355	,13725	4,2749	4,8679	4,00	5,00
Total	32	4,1875	,69270	,12245	3,9378	4,4372	2,00	5,00

Table E 25. Test of Homogeneity of Variances of Hypothesis 6

Levene Statistic	df1	df2	Sig.
1,358	2	29	,273

Table E 26. ANOVA of Hypothesis 6

	Sum of Squares	df	Mean Square	F	Sig.
Between Groups	3,696	2	1,848	4,795	,016
Within Groups	11,179	29	,385		
Total	14,875	31			

Table E 27. Robust Tests of Equality of Means(b) of Hypothesis 6

	Statistic(a)	df1	df2	Sig.
Welch	.	.	.	.
Brown-Forsythe	.	.	.	.

a Asymptotically F distributed.

b Robust tests of equality of means cannot be performed for Satisfaction_3 because at least one group has 0 variance.

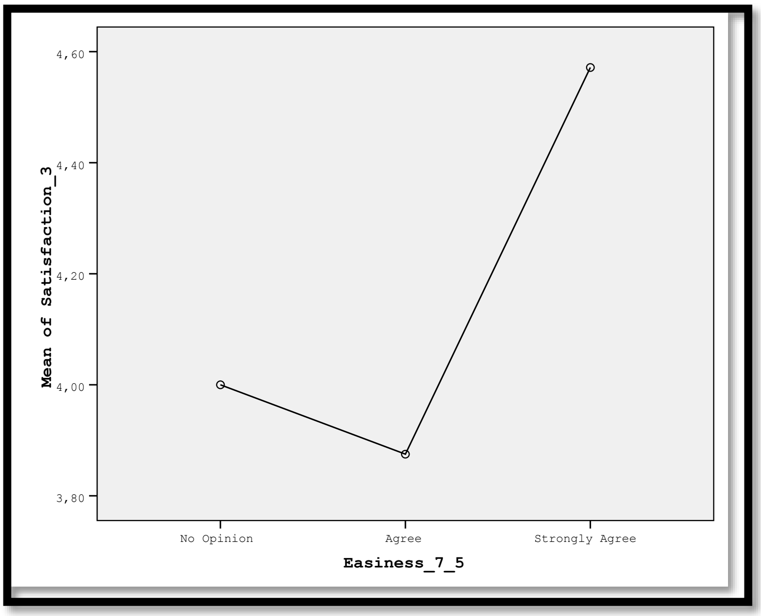


Figure E 6. Means Plots of Hypothesis 6

HYPOTHESIS 7

Table E 28. Descriptives of Hypothesis 7 that shows Satisfaction Level

	N	Mean	Std. Dev.	Std. Error	95% Confidence Interval for Mean		Min	Max
	Lower Bound	Upper Bound	Lower Bound	Upper Bound	Lower Bound	Upper Bound	Lower Bound	Upper Bound
Strongly Disagree	1	2,00	.	.	.	.	2	2
Disagree	2	5,00	,000	,000	5,00	5,00	5	5
No Opinion	10	3,80	,422	,133	3,50	4,10	3	4
Agree	15	4,33	,488	,126	4,06	4,60	4	5
Strongly Agree	4	4,75	,500	,250	3,95	5,55	4	5
Total	32	4,19	,693	,122	3,94	4,44	2	5

Table E 29. Test of Homogeneity of Variances of Hypothesis 7

Levene Statistic	df1	df2	Sig.
3,030(a)	3	27	,047

a Groups with only one case are ignored in computing the test of homogeneity of variance for Satisfaction_3.

Table E 30. ANOVA of Hypothesis 7

	Sum of Squares	df	Mean Square	F	Sig.
Between Groups	9,192	4	2,298	10,917	,000
Within Groups	5,683	27	,210		
Total	14,875	31			

Table E 31. Robust Tests of Equality of Means(b) of Hypothesis 7

	Statistic(a)	df1	df2	Sig.
Welch	.	.	.	.
Brown-Forsythe	.	.	.	.

a Asymptotically F distributed.

b Robust tests of equality of means cannot be performed for Satisfaction_3 because at least one group has the sum of case weights less than or equal to 1.

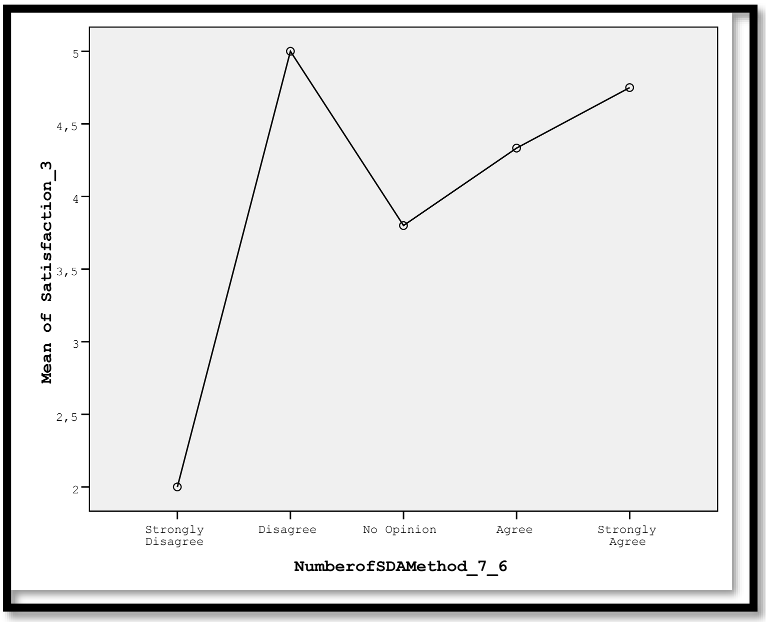


Figure E 7. Means Plots of Hypothesis 7

APPENDIX F

INSTALLATION OF THE sda4uDig

The steps and interface of installation is as follow:

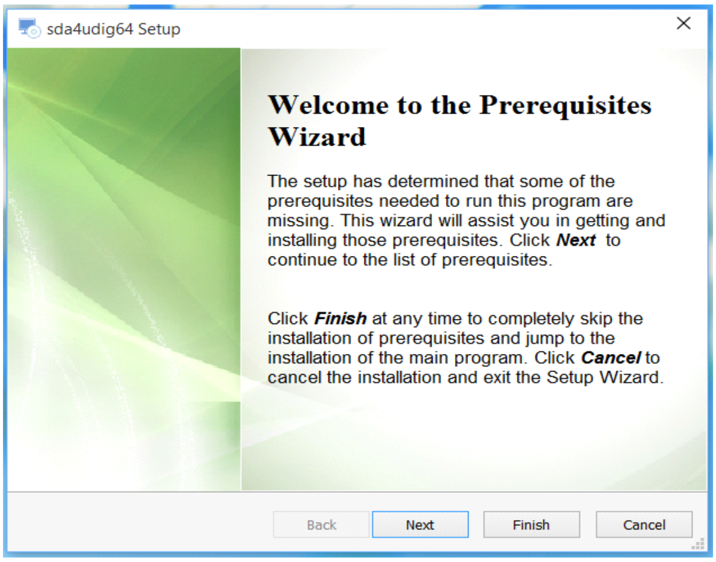


Figure F 1. Installation - Welcome Prerequisites Wizard

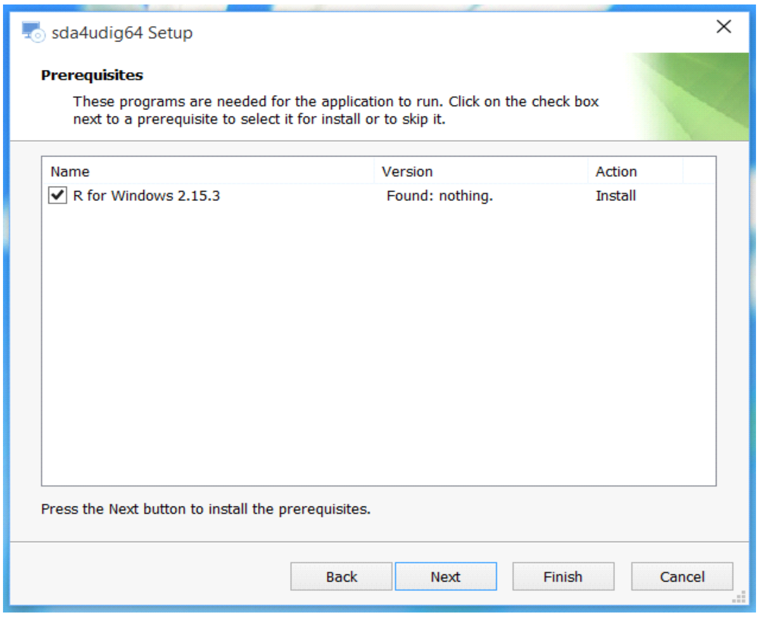


Figure F 2. Installation of R

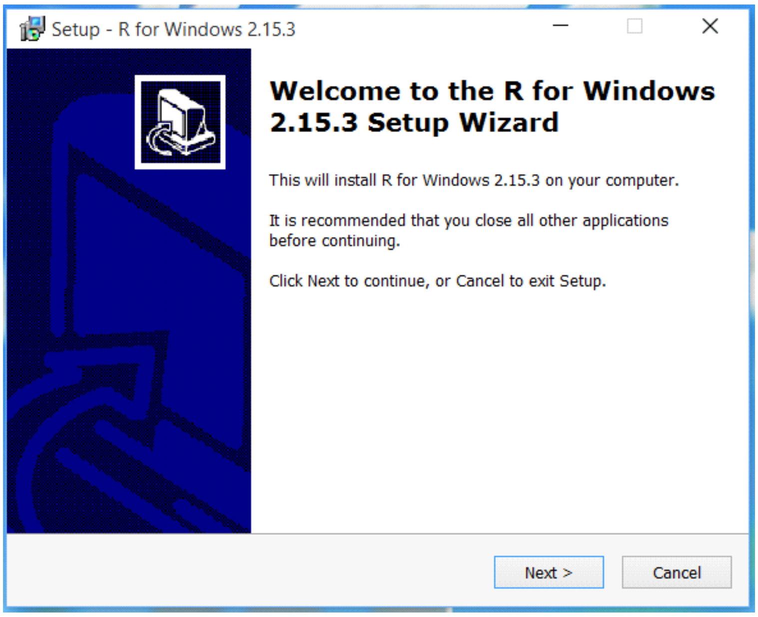


Figure F 3. Welcome Screen for R 2.15.3 Installation for Windows

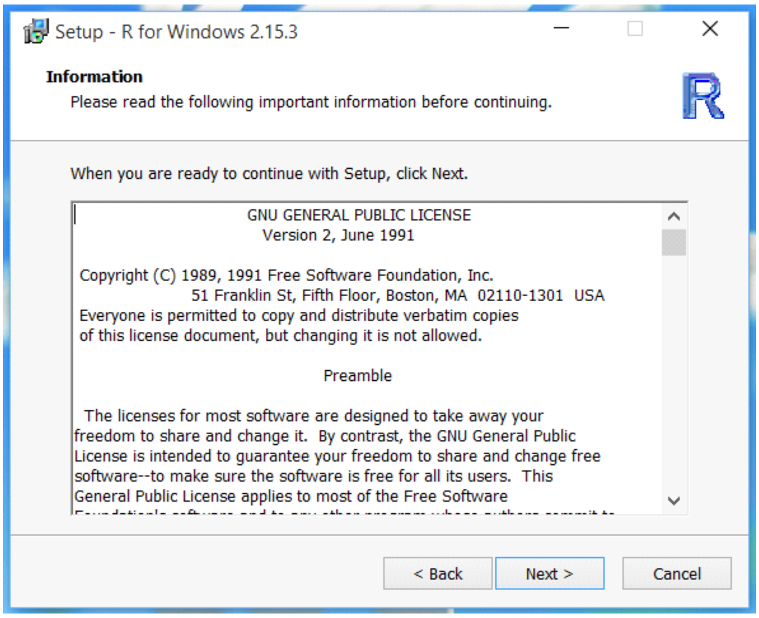


Figure F 4. Installation of R 2.15.3

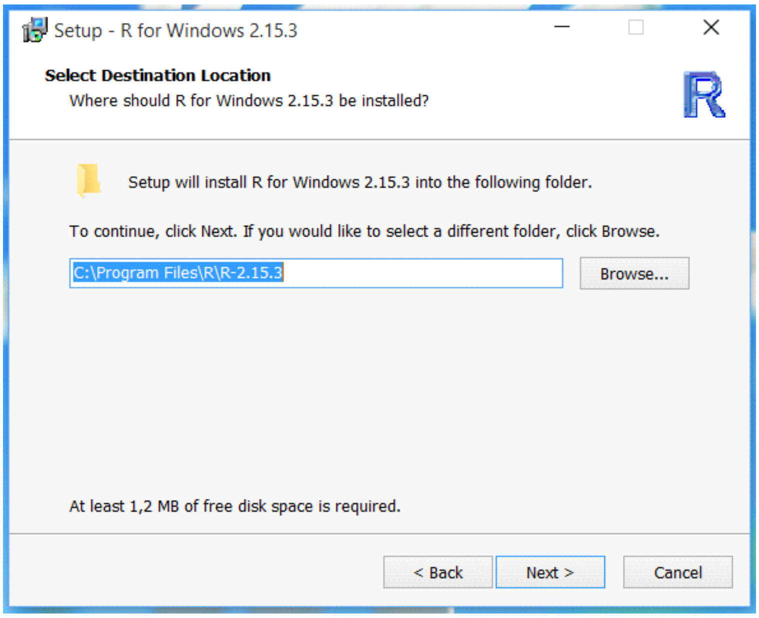


Figure F 5. Directory Definition

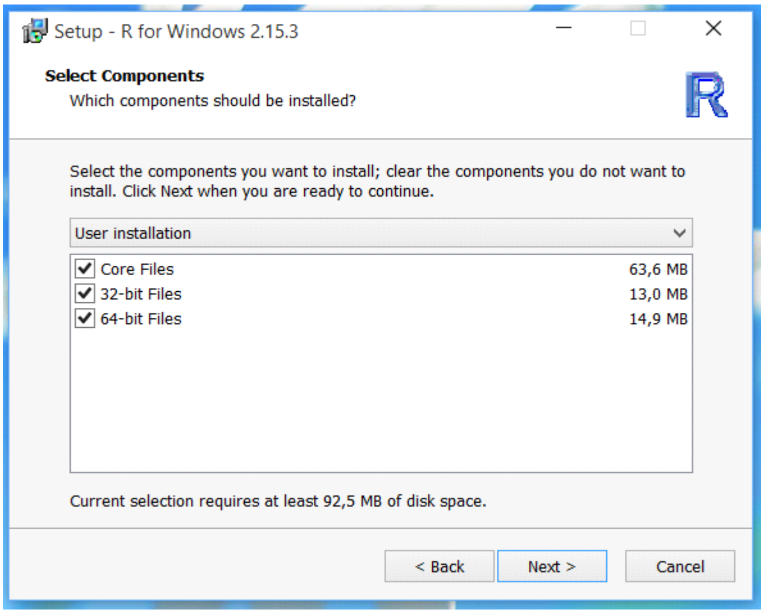


Figure F 6. OS Bit Selection

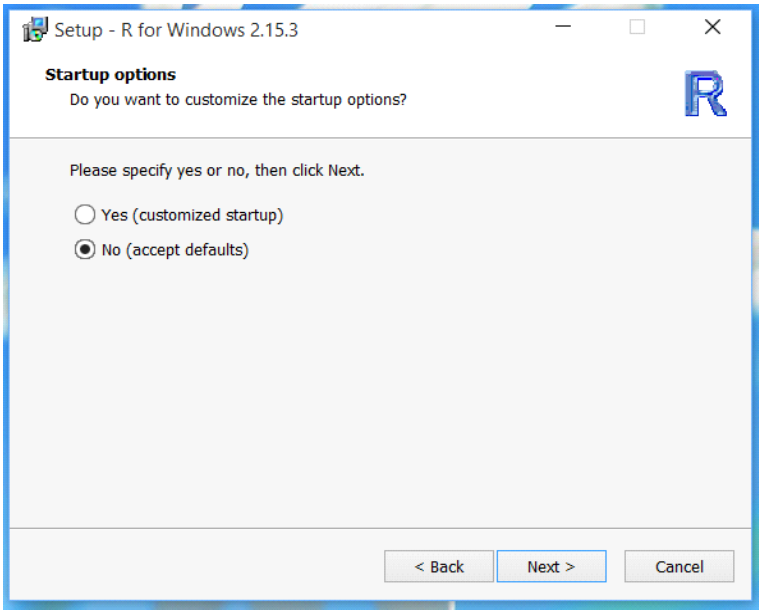


Figure F 7. Startup Options

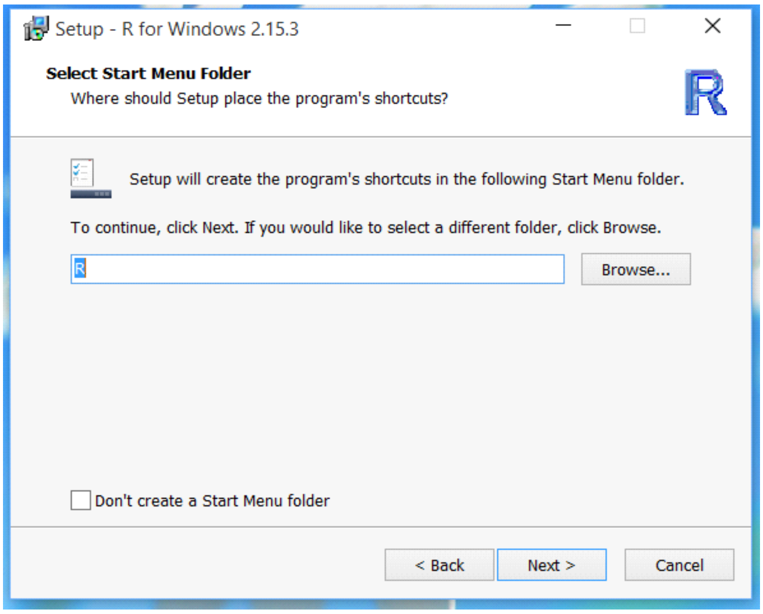


Figure F 8. Folder Selection for Setup

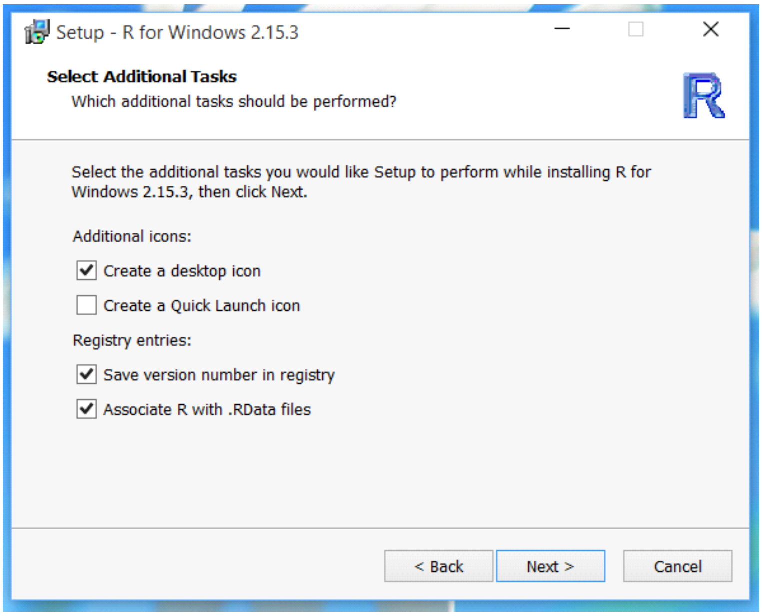


Figure F 9. Other Options for Setup

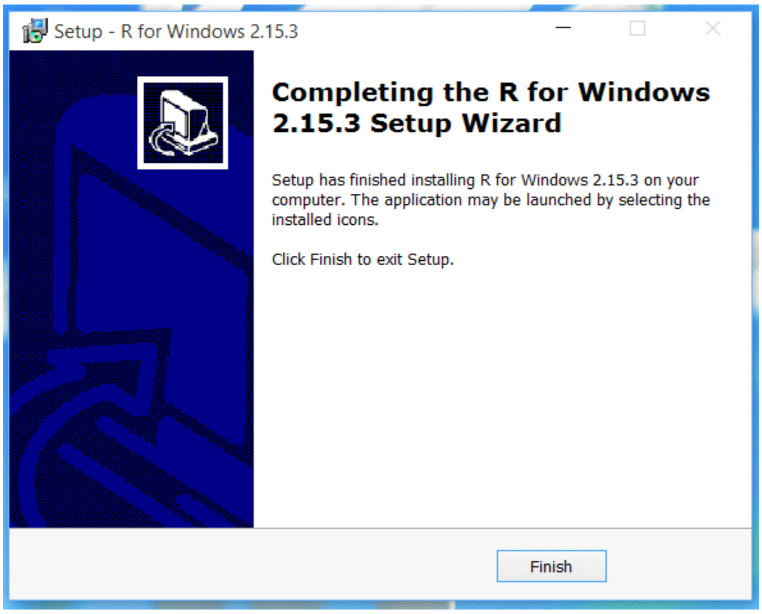


Figure F 10. Finish Installation Screen for R 2.15.3

After installing prerequities (R 2.15.3) of sda4uDig integration, the uDig GIS is installed with the sda4uDig Plug-in. Following figures show the steps of installation and their screenshots.

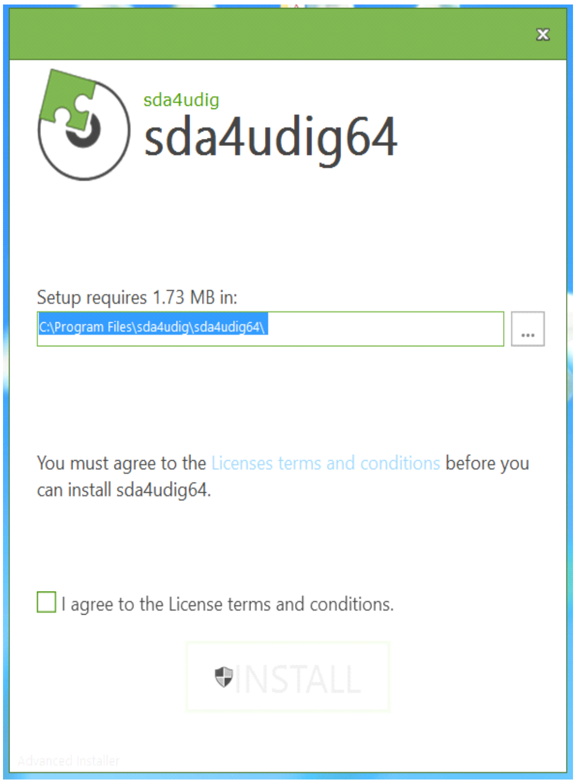


Figure F 11. Installation of sda4uDig for 64 Bit Win OS

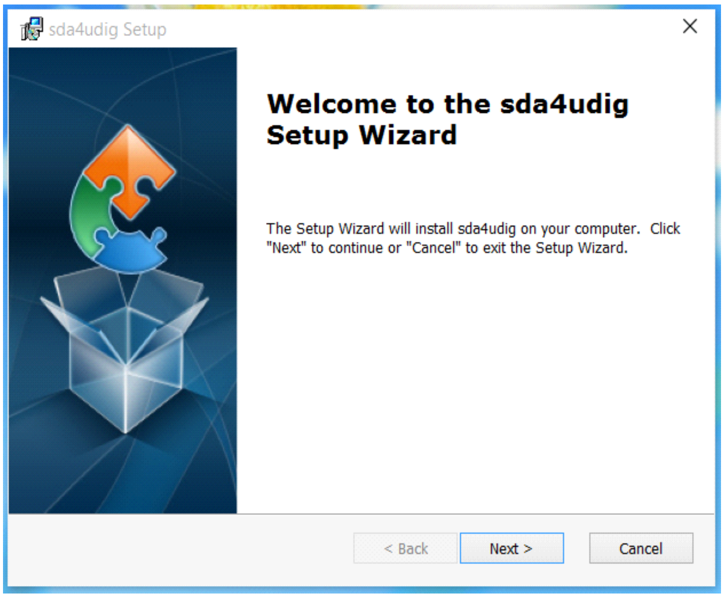


Figure F 12. Welcome to the sda4uDig Wizard

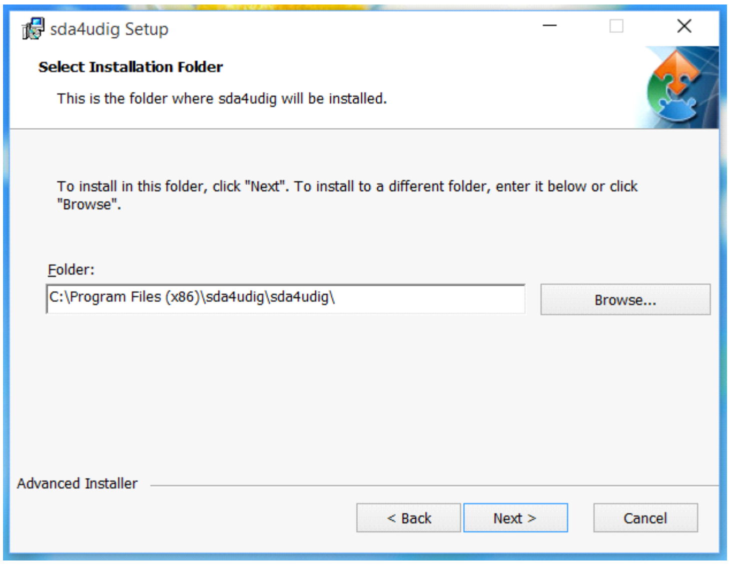


Figure F 13. Directory Selection

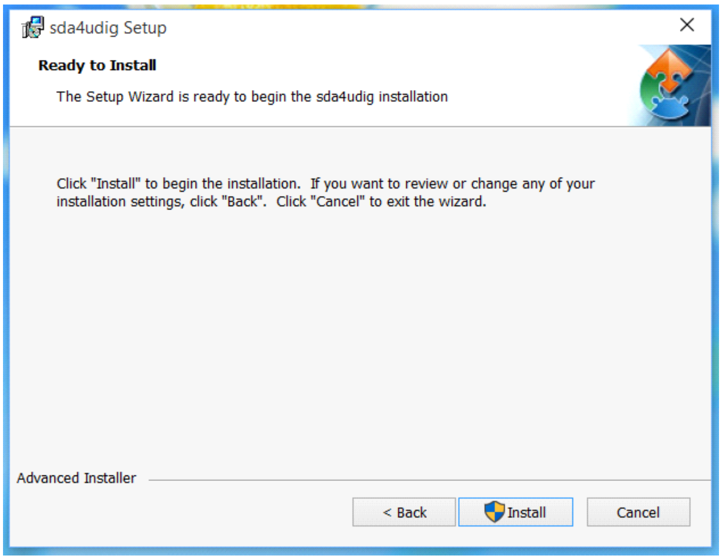


Figure F 14. Begin Installation

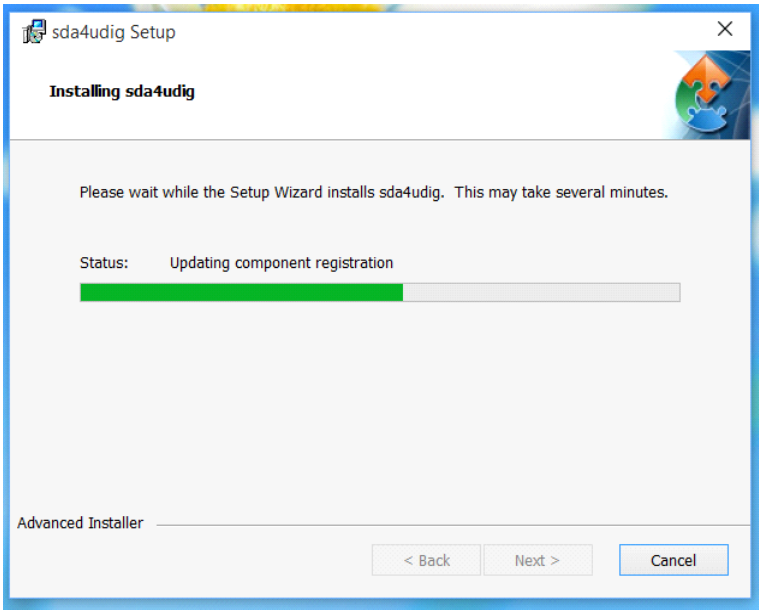


Figure F 15. Installing sda4uDig

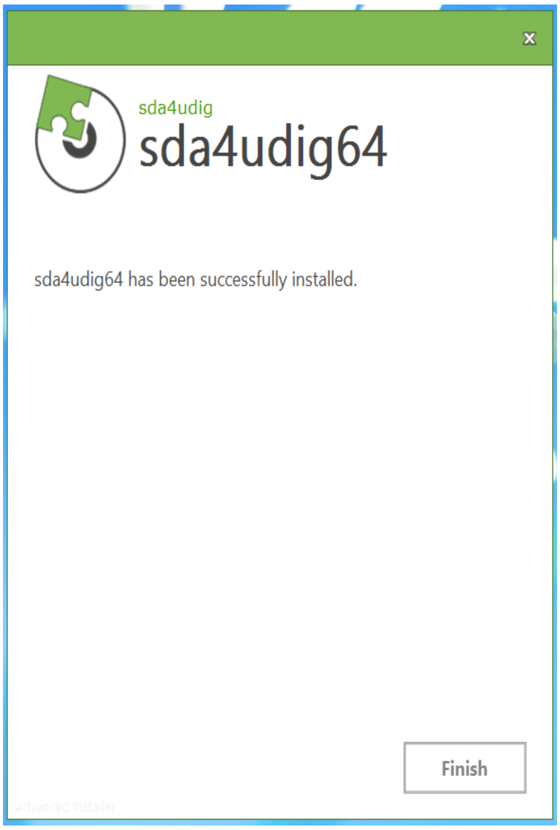


Figure F 16. Finish Installation of sda4uDig

Figure F 17 is designed and created an interface to complete the installation including R packages and some other configuraitons after completing installation of sda4uDig.

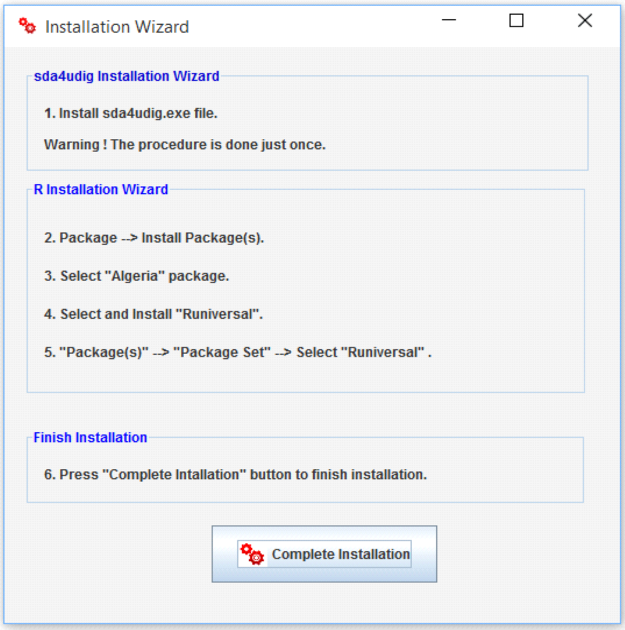


Figure F 17. Complete the R Packages Installation and Other Configuraitons

APPENDIX G

GEOSPATIAL AND NON-GEOSPATIAL MODULES INTEGRATED INTO uDig GIS

Figure G 1 shows the Geospatial Module including KDE, ADE, Uniform Intensity, Kriging, CSR, FPM, G-F-K-L Functions, Komolov Smirnov Test, PointinPoly Estimation, R Script Environment, 3D Visualizaiton.

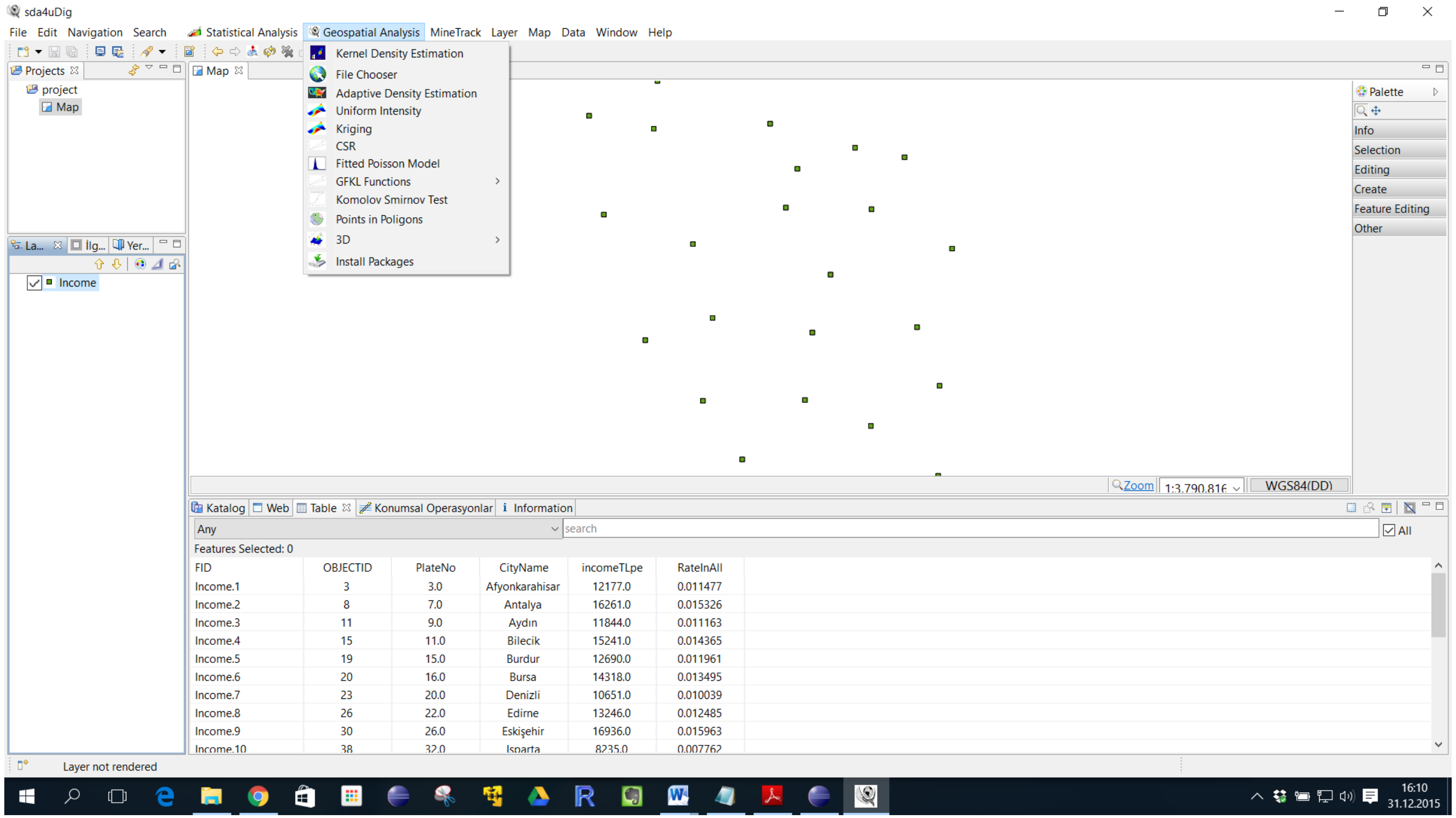


Figure G 1. Geospatial Module and Geospatial Techniques

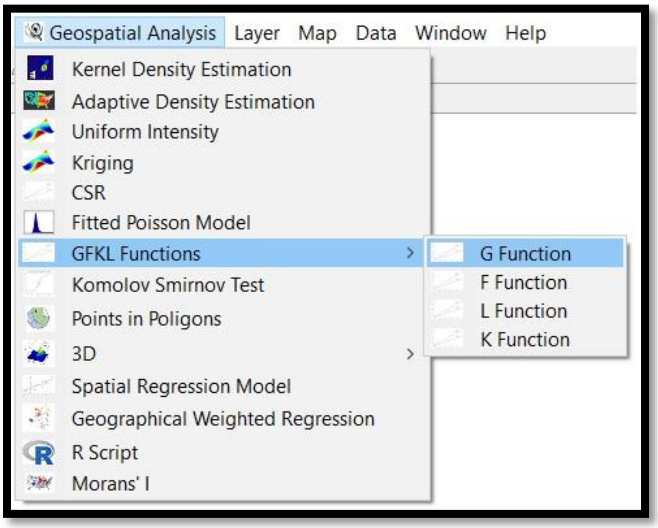


Figure G 1. (Continued)

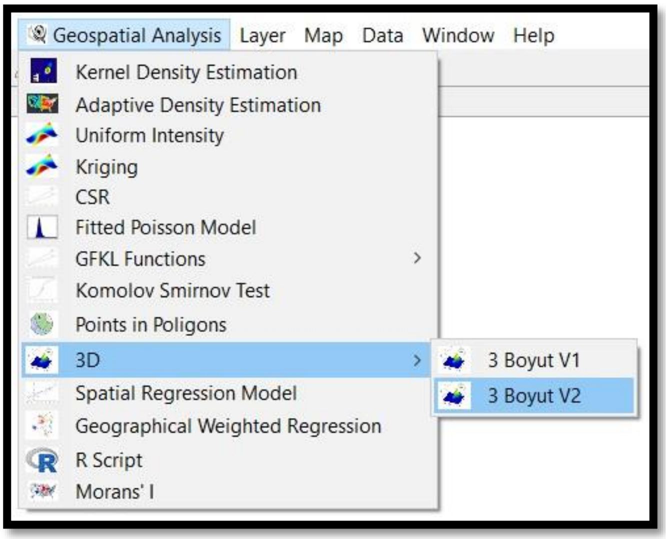


Figure G 1. (Continued)

Figure G 2 shows the Statistical Module including Column Graph, Pie Chart and Histogram Graph Menus.

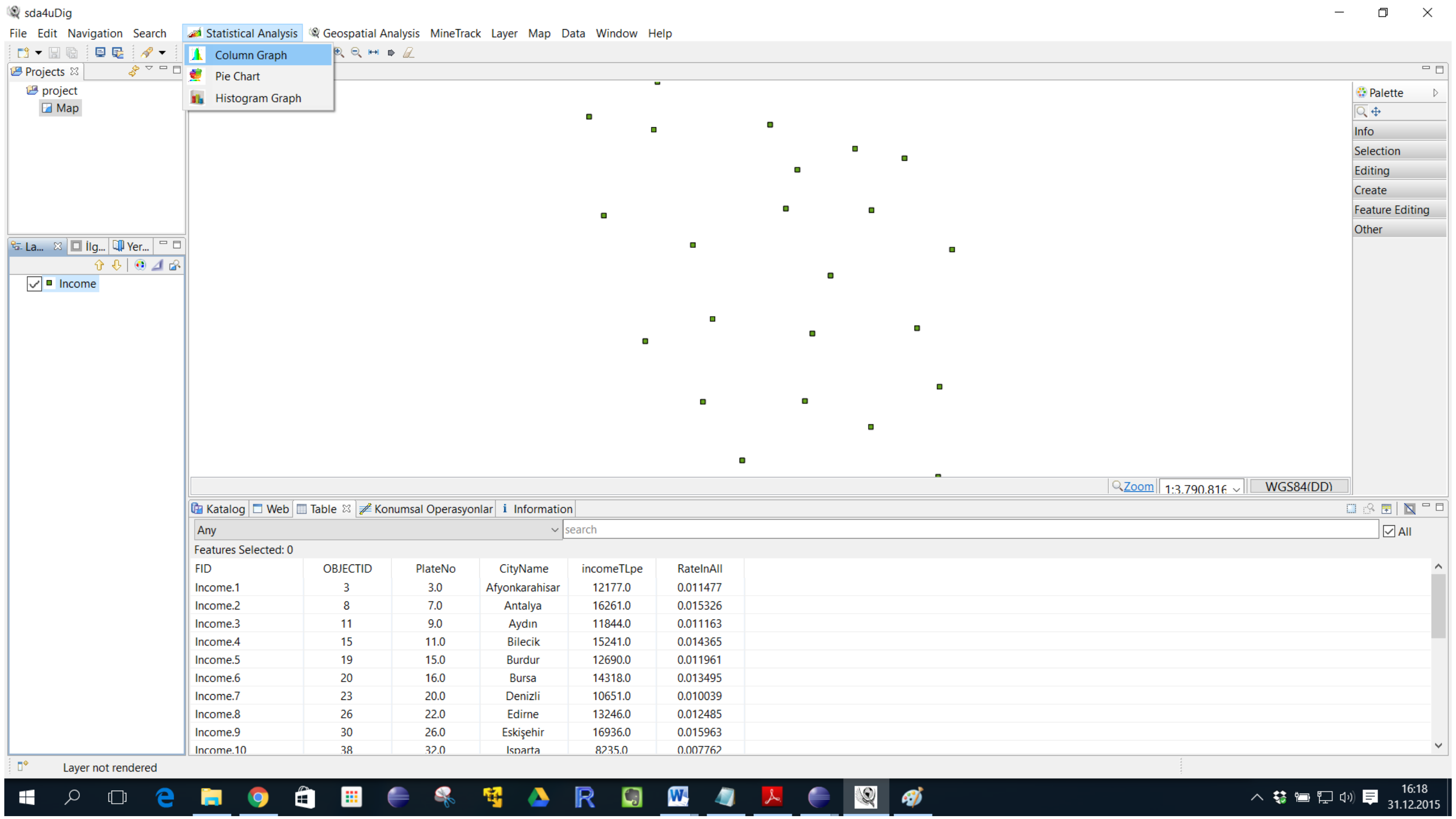


Figure G 2. Non-Geospatial Module and its Techniques

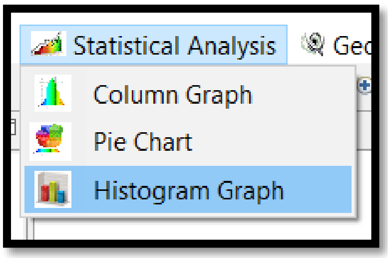


Figure G 2. (Continued)

Figure G 3 shows the Utility Module including several techniques.

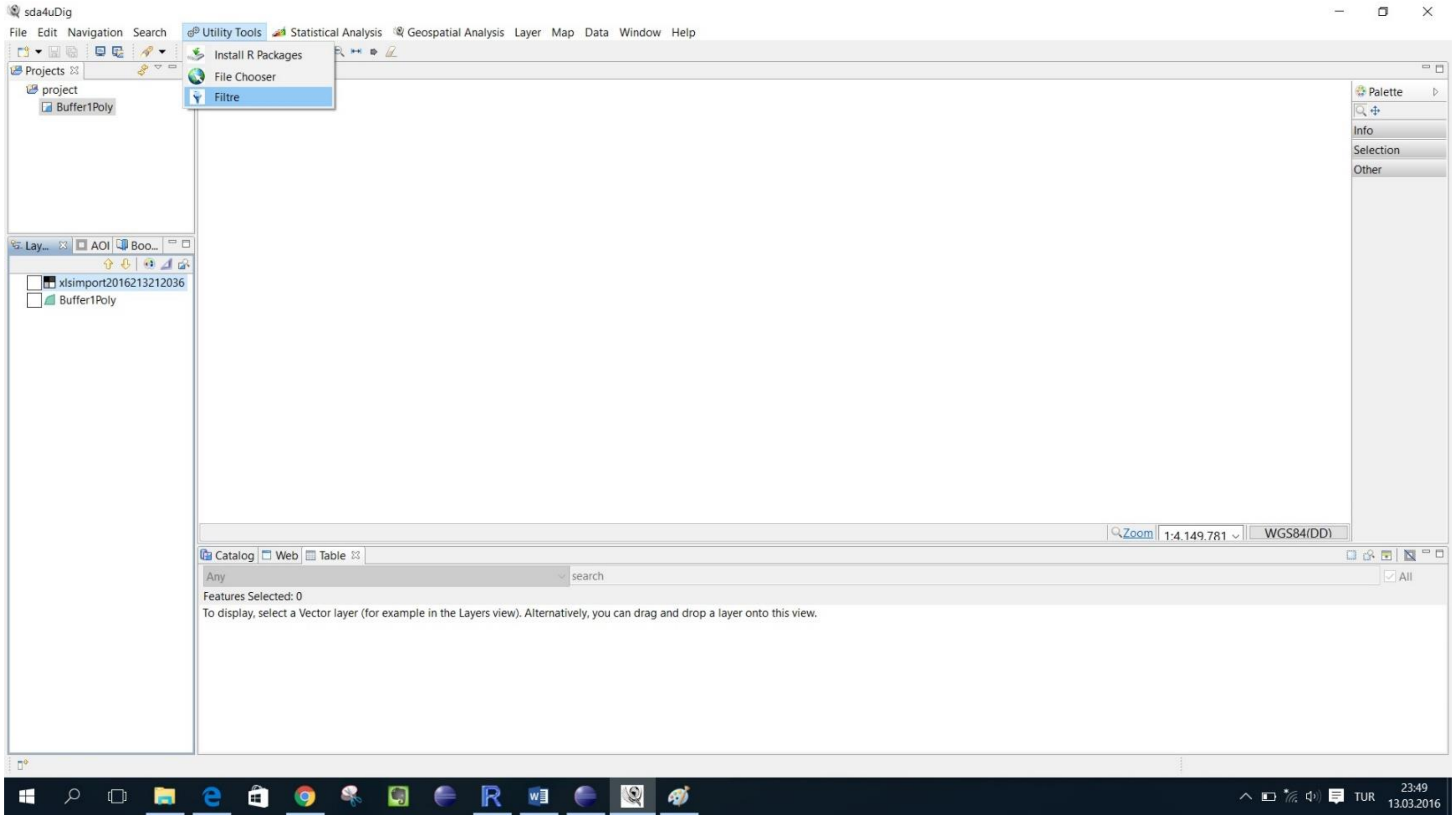


Figure G 3. Utility Module and its Techniques

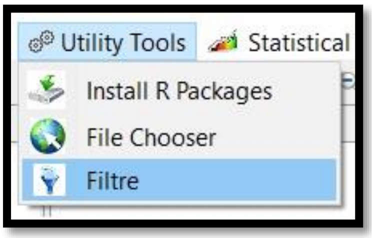


Figure G 3. (Continued)

APPENDIX H

INTEGRATED SDA TECHNIQUES, GUI AND ANALSYSIS RESULT

Adaptive Density Estimation

In Figure H 1, the input and output interface of ADE is shown.

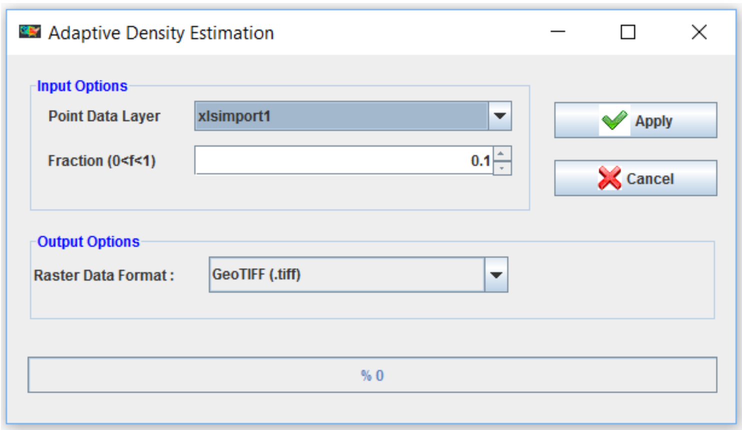


Figure H 1. ADE Iput/Output Interface

In Figure H 2, the result of ADE has been shown.

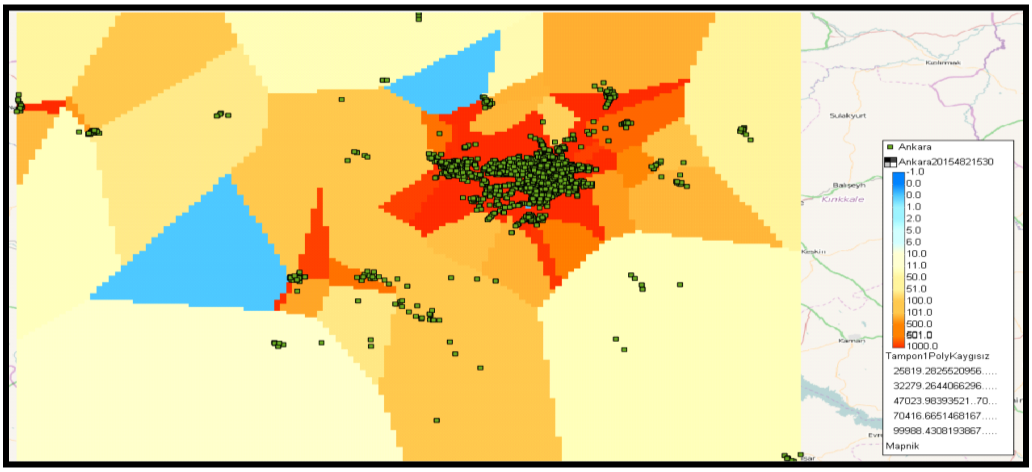


Figure H 2. ADE Result

Point/Area (Point in Poly) Density Method

In this method, points are buffered with respect to requirement using the “Spatial Operation” tool. Then all buffers are merged and the numbers of points in each buffer are calculated. Lastly, the density of each buffers are calculated by dividing the number of poins in each buffers to the merged buffers’ area. The final procedure is to classify the each buffer with respect to their density.

In Figure H 3, Spatial Operation interface has been shown for buffer operation.

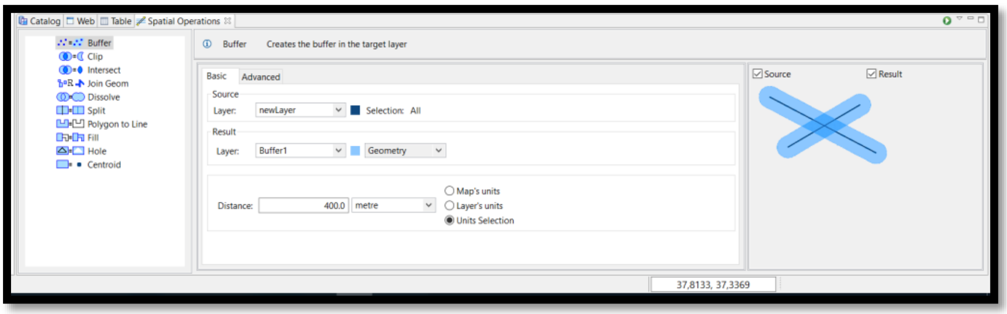


Figure H 3. Spatial Operation Interface

In Figure H 4 is shown. It adds new three colomns which are “Number of Points”, “Area of Buffers” and “Density of Buffers” in the shape data. Then, it calculates each added colomns and writes in the shape (.dbf) file. It calculates each added colomns and writes in the shape (.dbf) file. It calculates each added colomns and writes in the shape (.dbf) file. It calculates each added colomns and writes in the shape (.dbf) file.

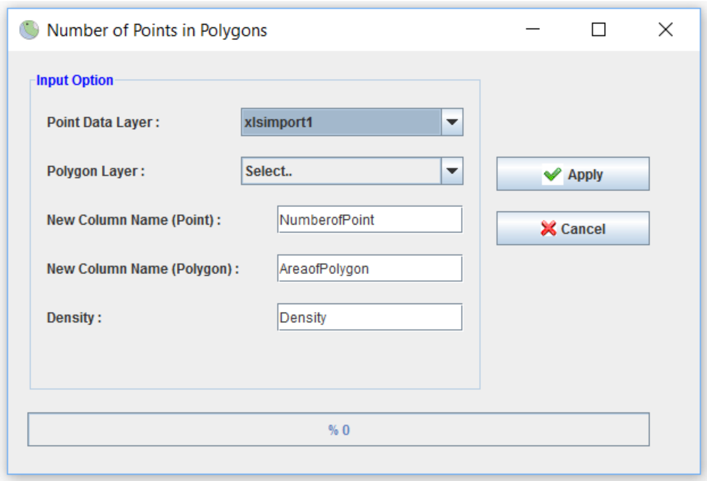
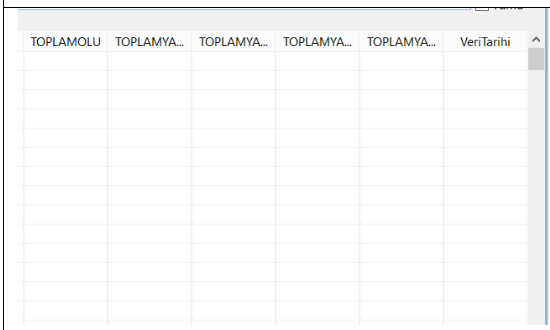
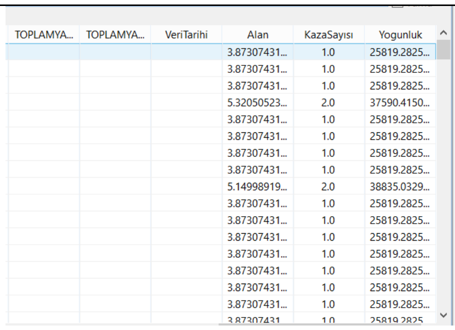


Figure H 4. Point in Poly Method I/O Interface

Table H 1. Dbf file of Data Before and After PointinPoly Method

<table>
<tr><td>Before Point in Poly Method</td><td>After Point in Poly Method Applied</td></tr>
<tr><td>
</td><td>
</td></tr>
<tr><td></td><td>Three new columns have been added</td></tr>
<tr><td colspan="2">It calculates three new column’s area, number of points in poligons and density and adds them into dbf file.</td></tr>
</table>

In Figure H 5, the result of PointinPoly or Density Estimation Method has been shown.

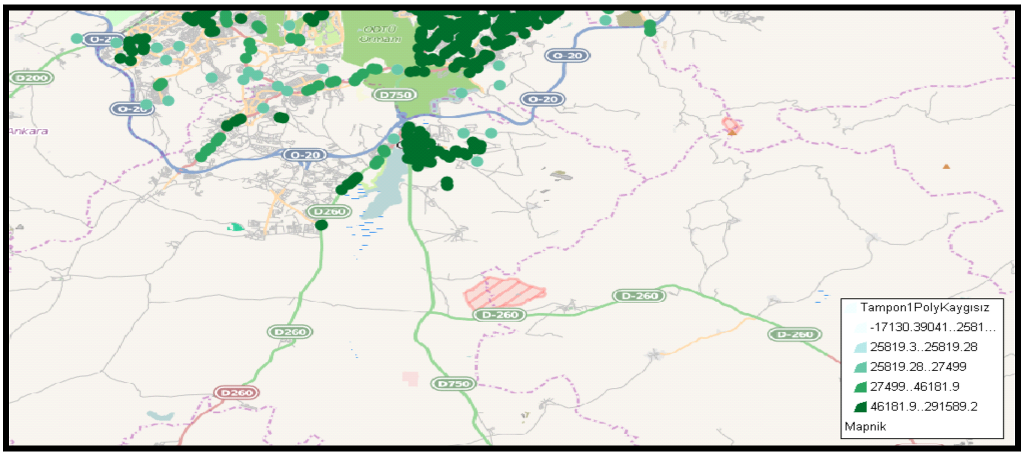


Figure H 5. PointinPoly Result

Simulation Envelope for CSR

In Figure H 6, the input and output interface of CSR is shown.

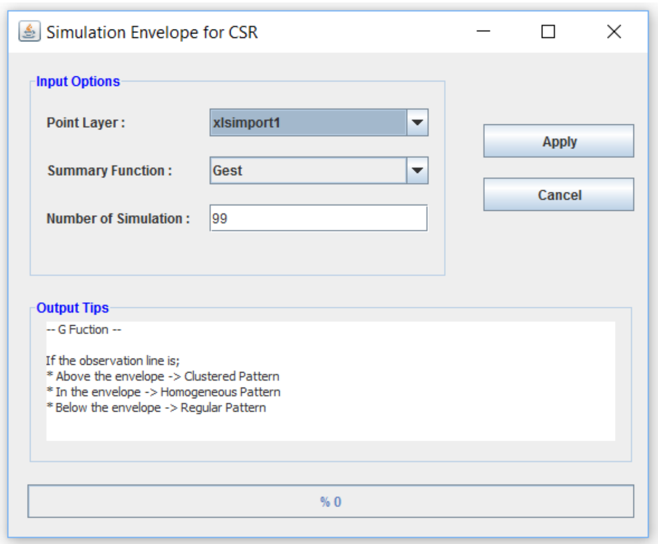


Figure H 6. CSR Interface I/O Interface

In Figure H 7, the result of CSR using Gest function is shown. There are other functions that can be applied such as Kest, Fest and Lest.

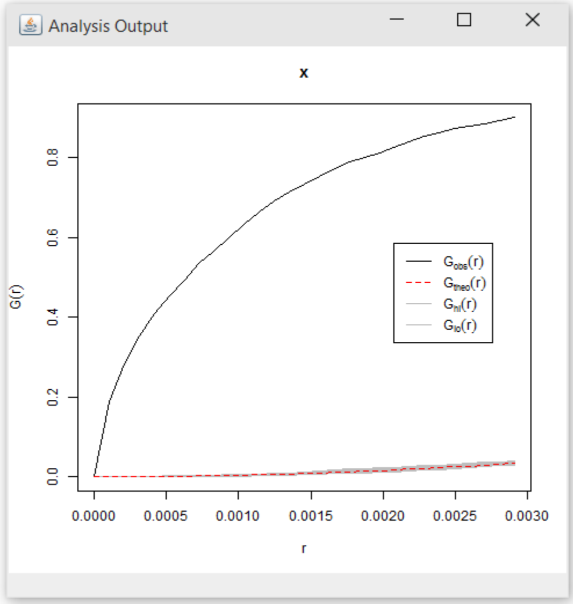


Figure H 7. Resulf of CSR

K Function

In Figure H 8, the input and output interface of K Estimation is shown.

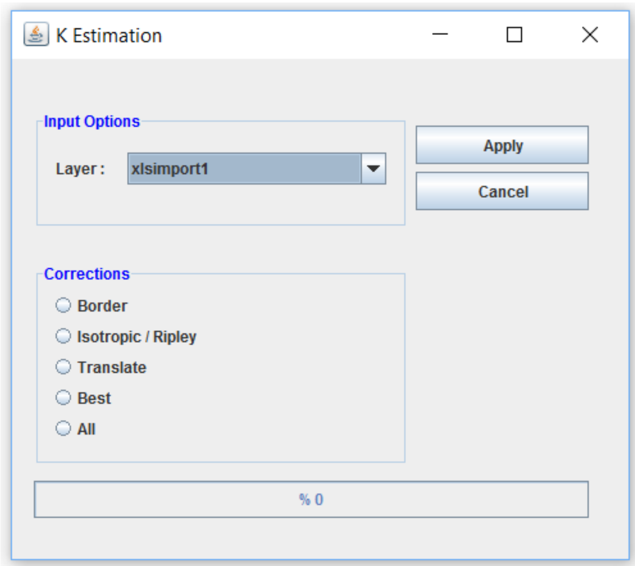


Figure H 8. K Estimation I/O Interface

In Figure H 9, the result of K Estimation using All Choice has been shown.

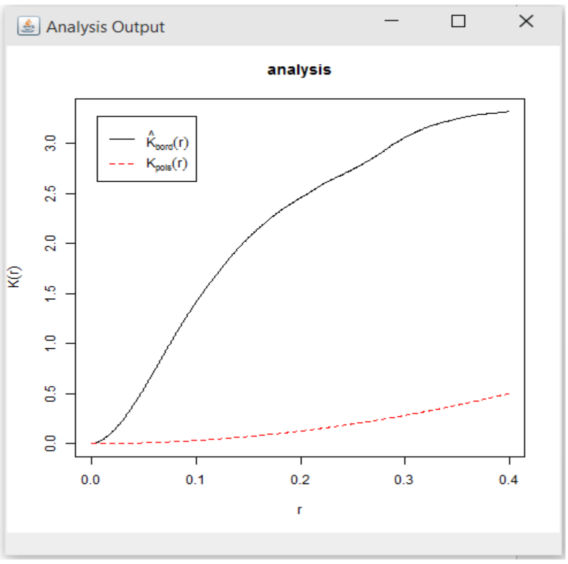


Figure H 9. K Estimation Result

L Function

In Figure H 10, the input and output interface of L Estimation is shown.

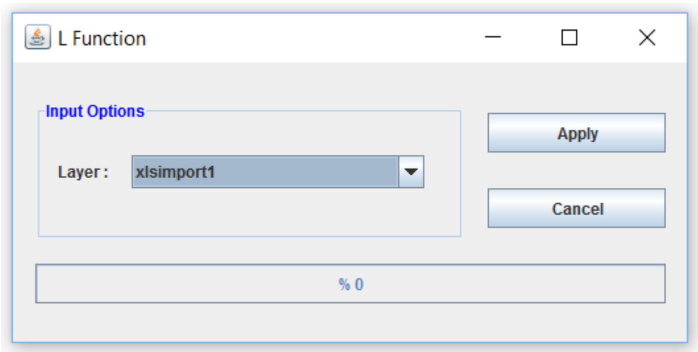


Figure H 10. L Estimation I/O Interface

In Figure H 11, the result of L Estimation using All Choice is shown.

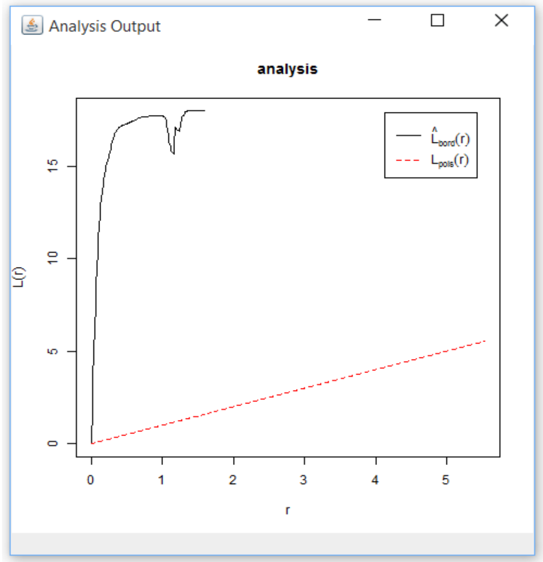


Figure H 11. L Estimation Result

G Function

In Figure H 12, the input and output interface of G Estimation is shown.

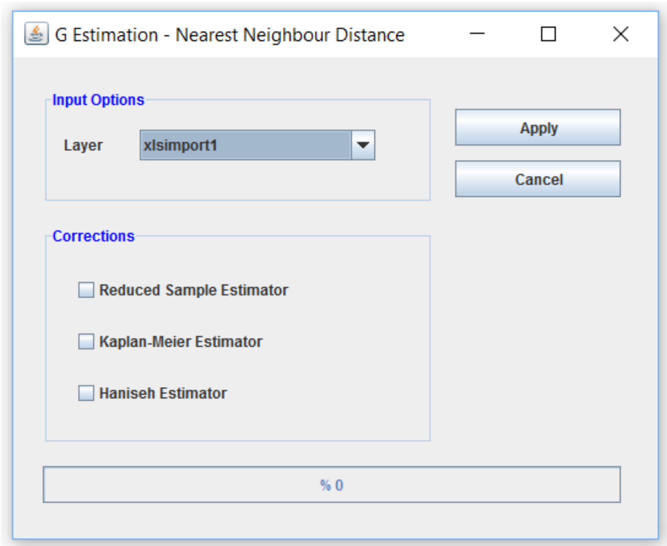


Figure H 12. G Estimation I/O Interface

In Figure H 13, the result of G Estimation using Haniseh Estimator is shown.

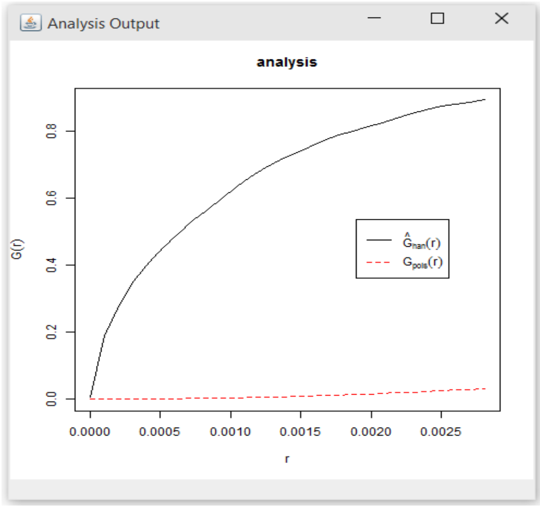


Figure H 13. G Estimation Result

F Function

In Figure H 14, the input and output interface of F Estimation is shown.

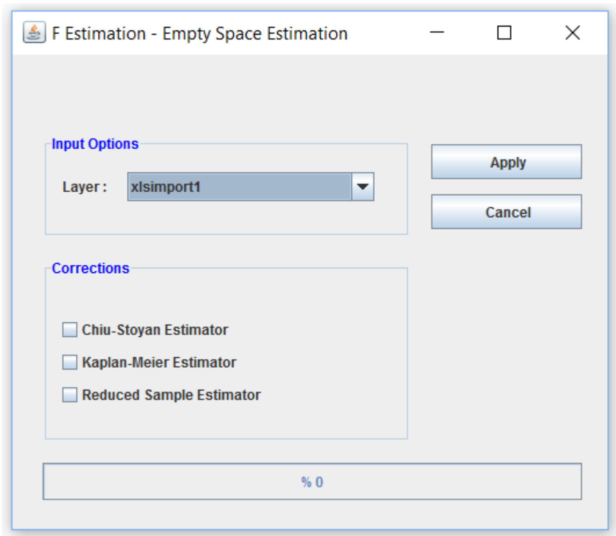


Figure H 14. F Estimation I/O Interface

In Figure H 15, the result of F Estimation using Chiu-Stoyan Estimator is shown.

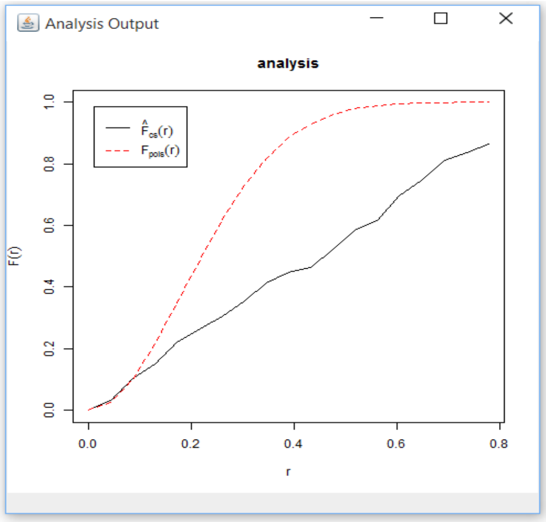


Figure H 15. F Estimation Result

Komolov Smirnov Test

In Figure H 16, the input and output interface of KST is shown.

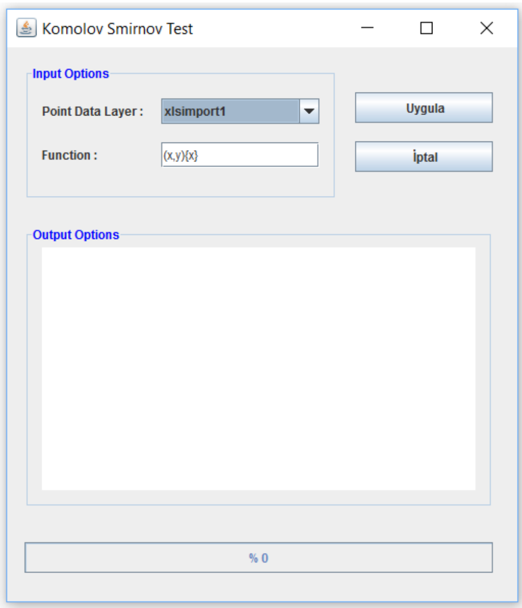


Figure H 16. KST I/O Interface

In Figure H 17, the result of KST is shown.

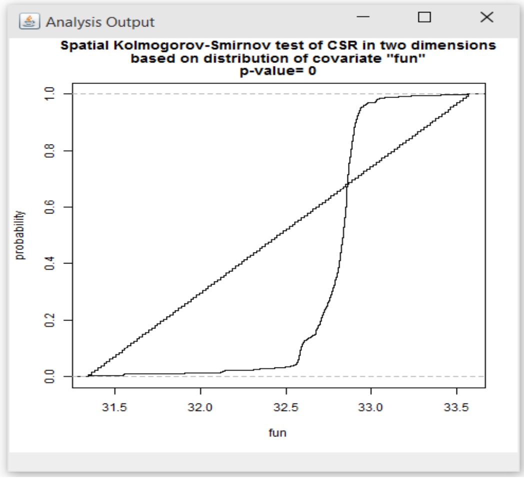


Figure H 17. KST Result

Fit Poisson Test

In Figure H 18 is shown.

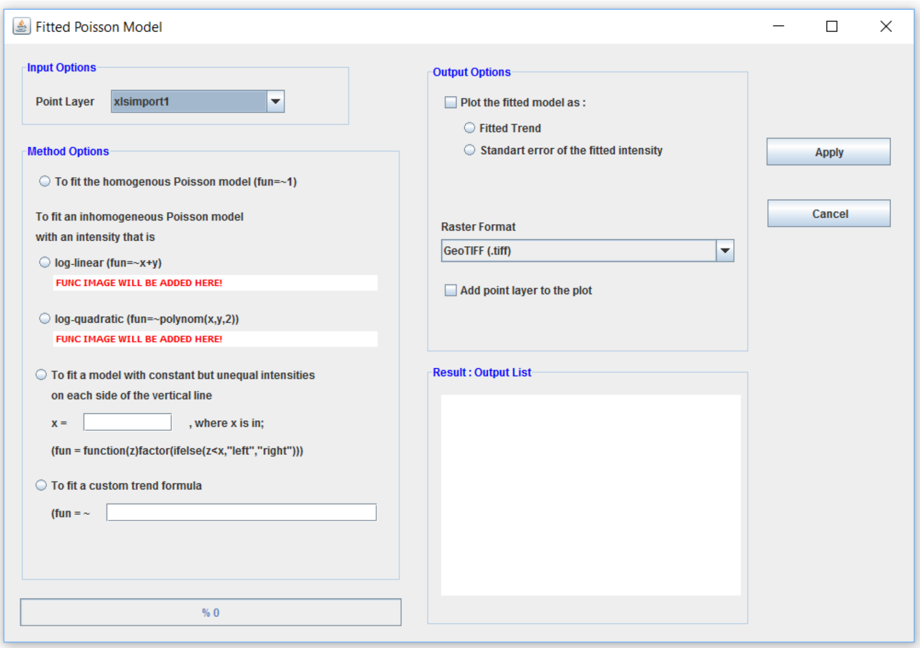


Figure H 18. FPM I/O Interface

In Figure H 19 and Figure H 20, the result of FPM using log quadratic model is shown.

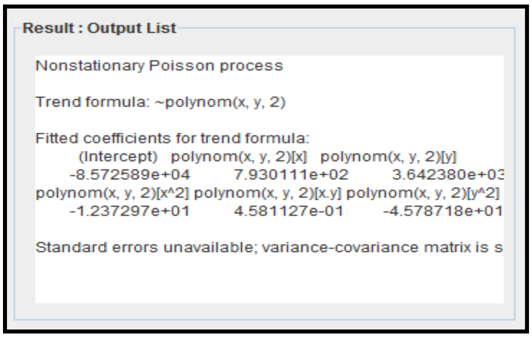


Figure H 19. FPM Result on I/O Interface

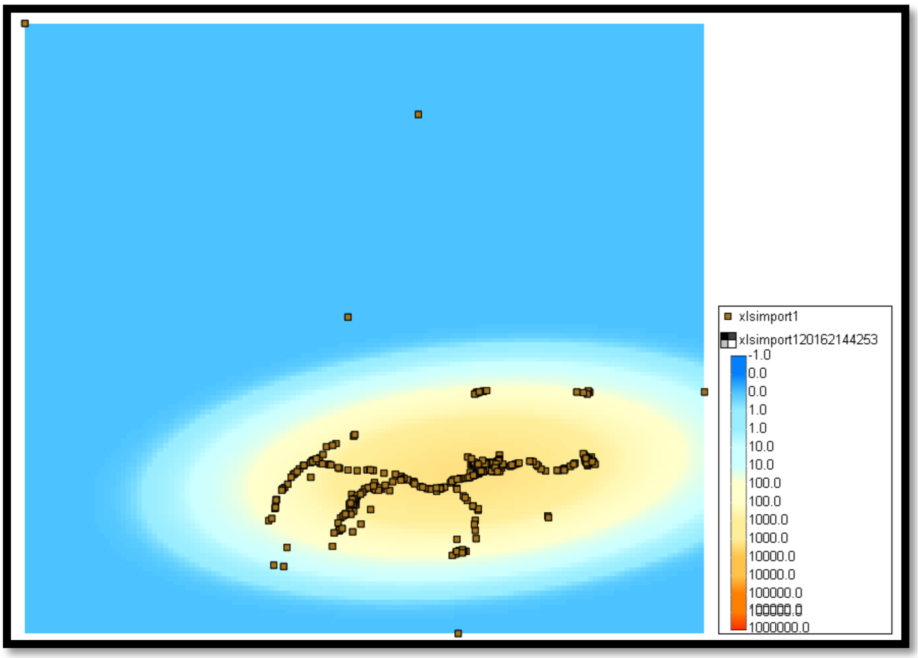


Figure H 20. FPM Result

Uniform Intensity

In Figure H 21, the input and output interface of Uniform Intensity is shown.

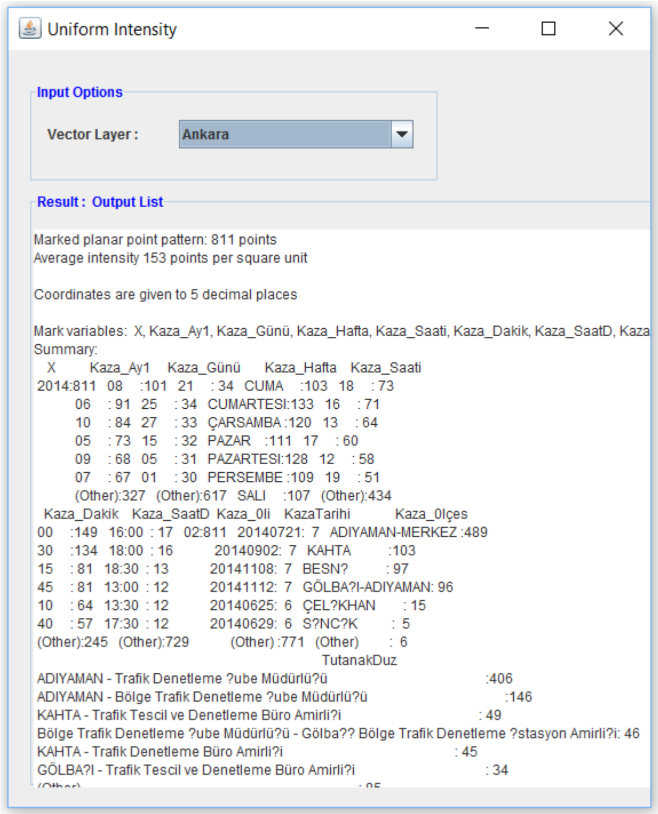


Figure H 21. Uniform Intensity I/O Interface

Spatial Regression Model

In Figure H 22, the input and output interface of Spatial Regression Model is shown.

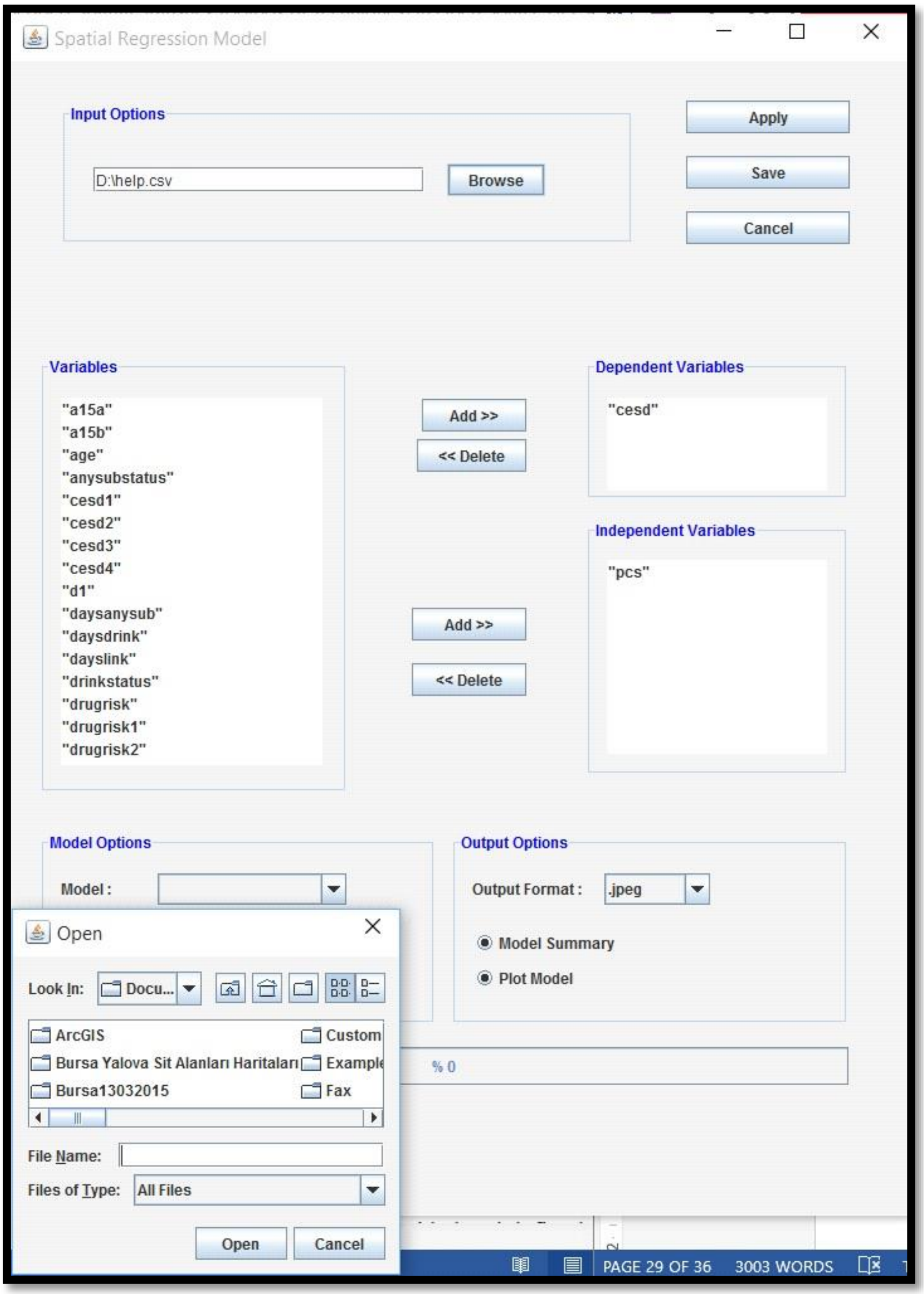


Figure H 22. SRM I/O Interface

In Figure H 23, the result of SRM is shown.

Figure H 23. Spatial Regression Model Result

<u>Geographical Weighted Regression Model (GWR)</u>

In Figure H 24, the input and output interface of Geographical Regression Model is shown.

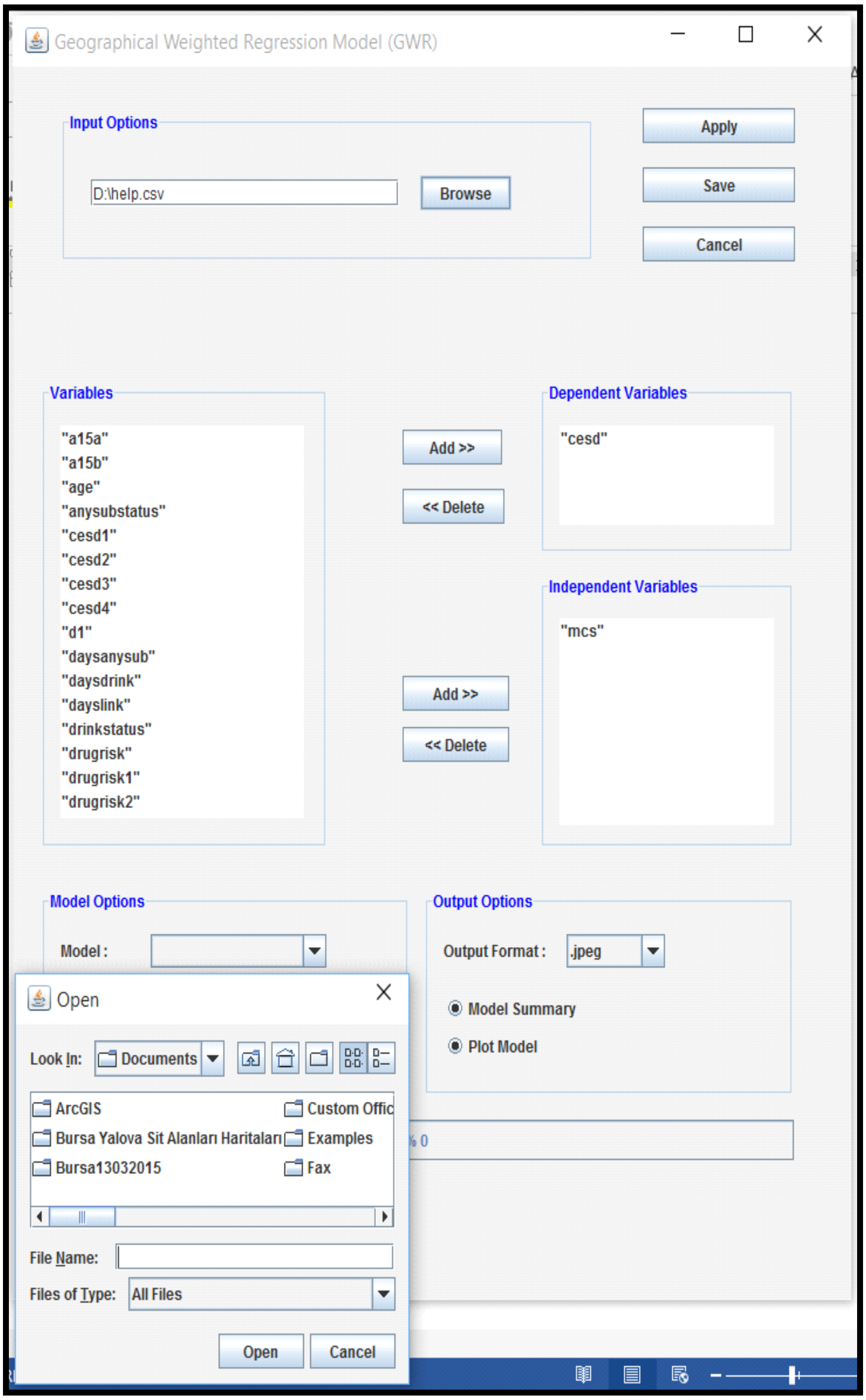


Figure H 24. GWR Model I/O Interface

In Figure H 25, the result of GWR is shown.

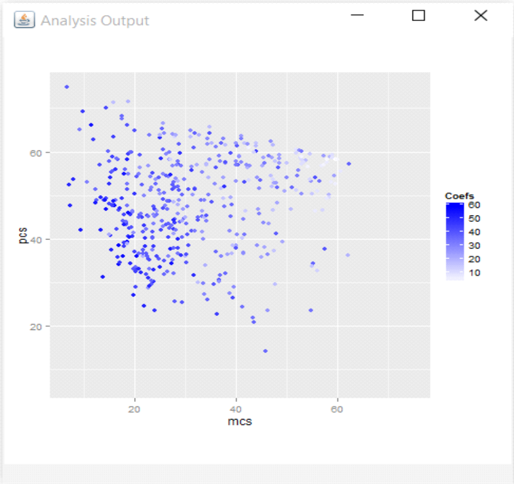


Figure H 25. Geographical Regression Model (GWR) Result

<u>Kriging</u>

In Figure H 26, the input and output interface of Kriging Model is shown.

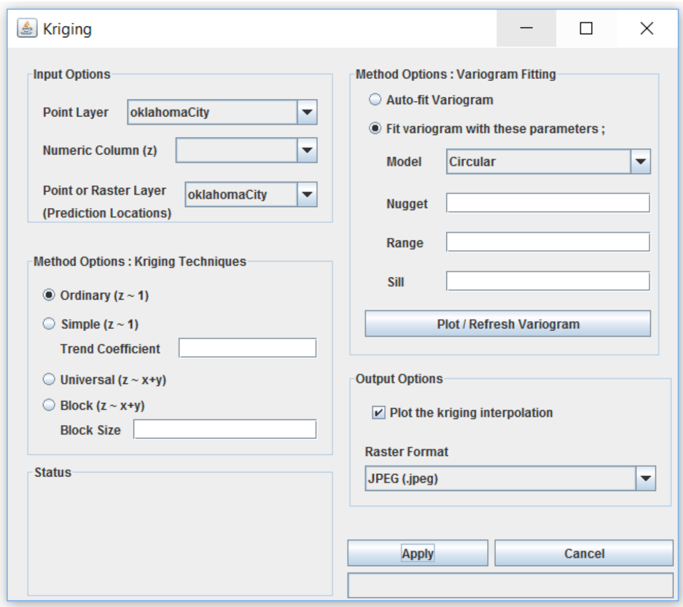


Figure H 26. Kriging Model I/O Interface

In Figure H 27, the result of variogram is shown.

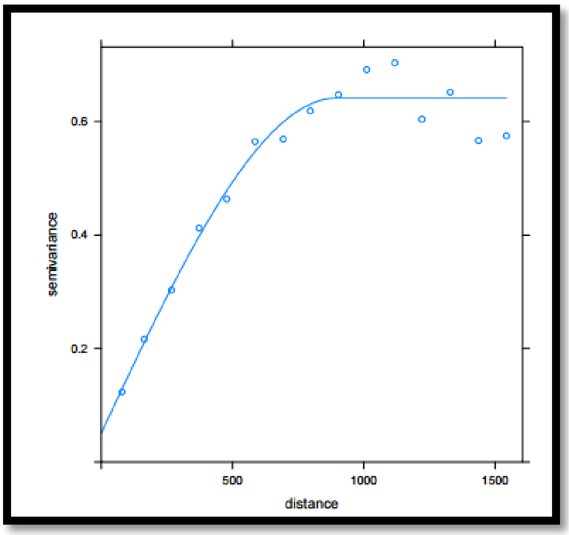


Figure H 27. Result of Kriging Model- Semivariogram

Figure H 28 shows the ordinary kriging result.

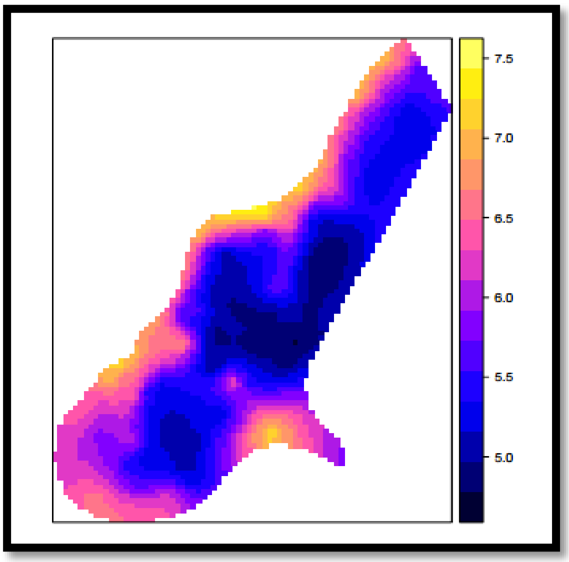


Figure H 28. Result of Kriging Model – Ordinary Kriging Applied

In addition to geospatial techniques, there are some statistical (non-geospatial) and visualization techniques have been embedded into uDig. These techniques are Colomn Graph, Histogram Graph and Pie Chart for statistical summary. Also, there are two different 3D visualization choices have been integrated to GIS interface. Moreover, the csv, excel and mdb type of input style have been added to uDig. Finally, in order to filter the data to make analysis with a part of data, the filter dialog has been added to uDig.

Histogram Graph

In Figure H 29, the input and output interface of Histogram has been shown.

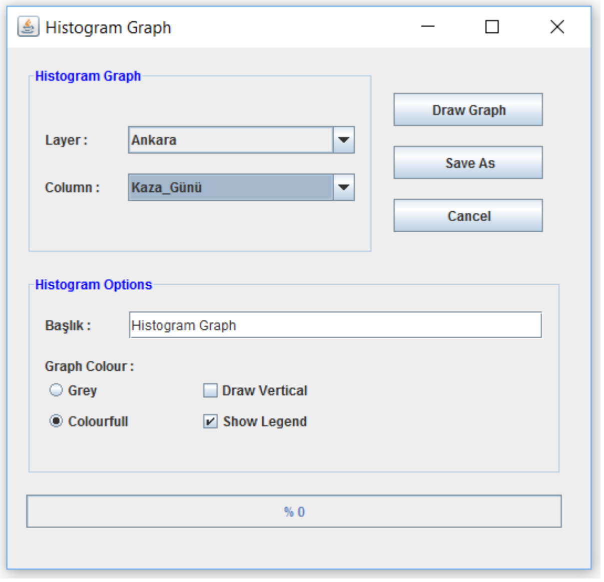


Figure H 29. Histogram Graph I/O Interface

In Figure H 30, the result of Histogram shows that the number of dead in accidents with respect to weekdays.

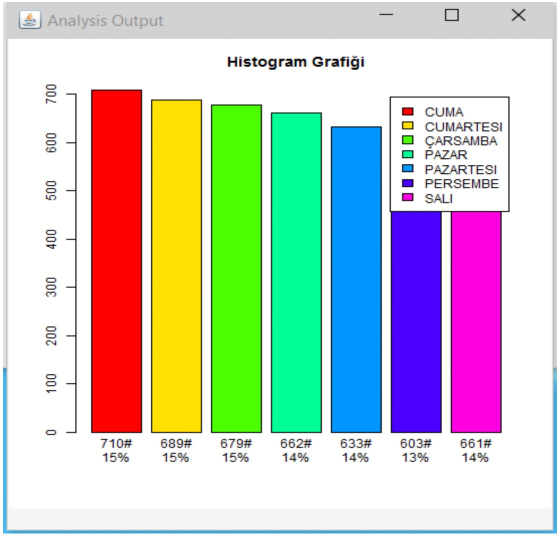


Figure H 30. Histogram Graph Result

Pie Chart

In Figure H 31, Pie Chart non-geopspatial technique GUI is shown.

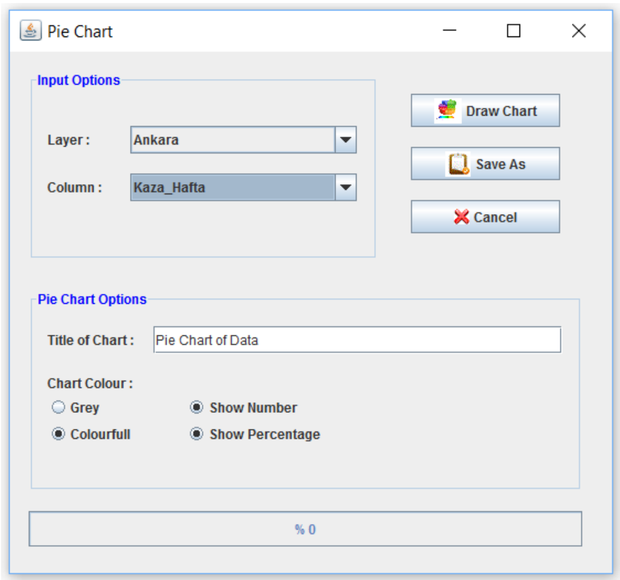


Figure H 31. Pie Chart I/O Interface

In Figure H 32, the result of Pie Chart shows that the number of dead in accidents with respect to weekdays.

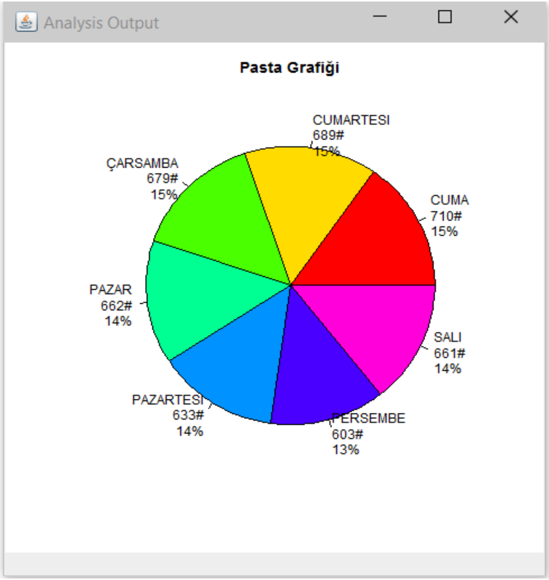


Figure H 32. Pie Chart Result

Column Chart

In Figure H 33, the input and output interface of Column Chart is shown.

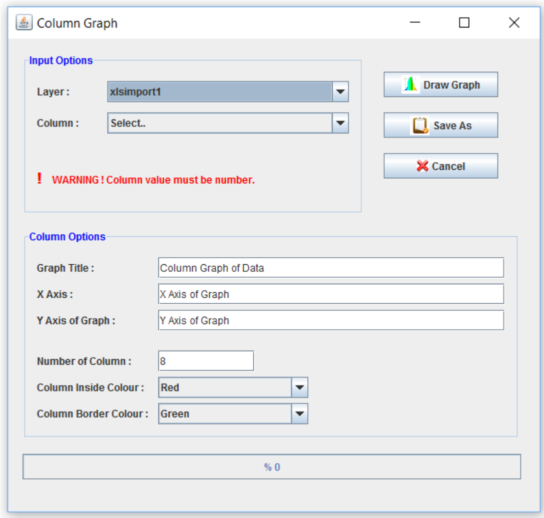


Figure H 33. Column Graph I/O Interface

In Figure H 34, the result of Column Chart shows that the number of dead in accidents with respect to weekdays.

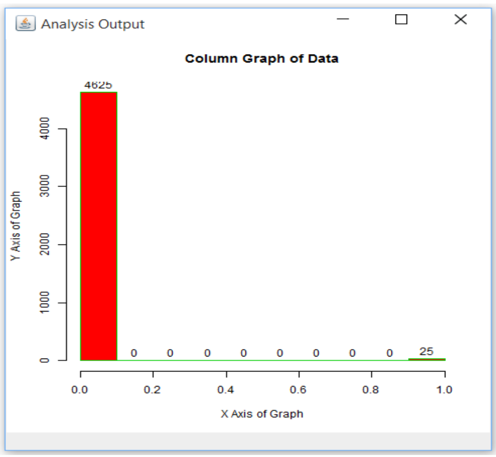


Figure H 34. Pie Chart Result

3D Visualization of Analysis Result

There are two different 3D visualizations are shown in the following figures. In Figure H 35, the input and output interface of 3D V1 is shown.

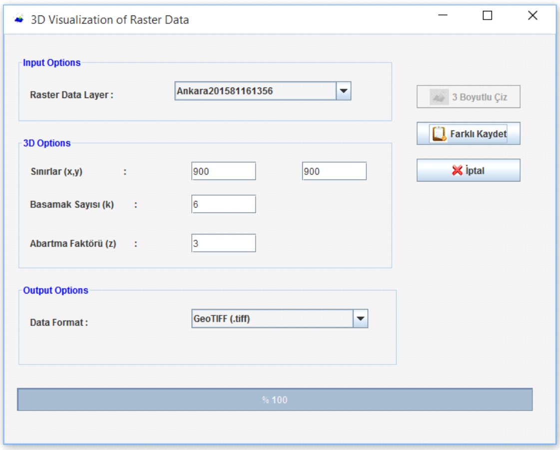


Figure H 35. 3D Volume1 I/O Interface

In Figure H 36, the result of 3D V1 using KDE is shown.

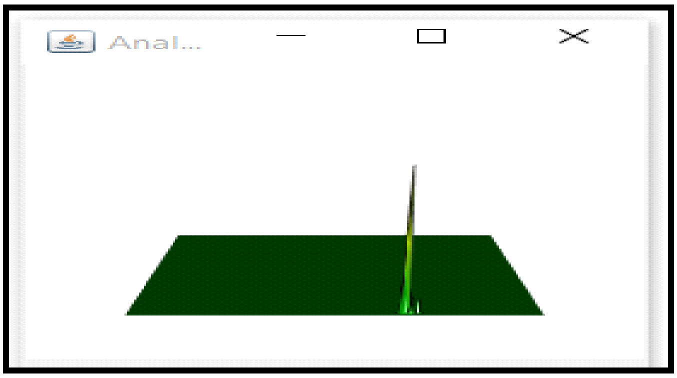


Figure H 36. 3D V1 Result

As before pointed out that, the user have different type of data in hand and they want to use these data in GIS softwares. In this part, there are three different data types have been embedded into uDig GIS software other than shape file. These data formats are ASCII, excel and mdb which are shown in the following figures. In Figure H 37, the input interface of Excel, Csv and Mdb is shown. All tree input style are almost working in the same idea, therefore, the usage of one of them is explained here.

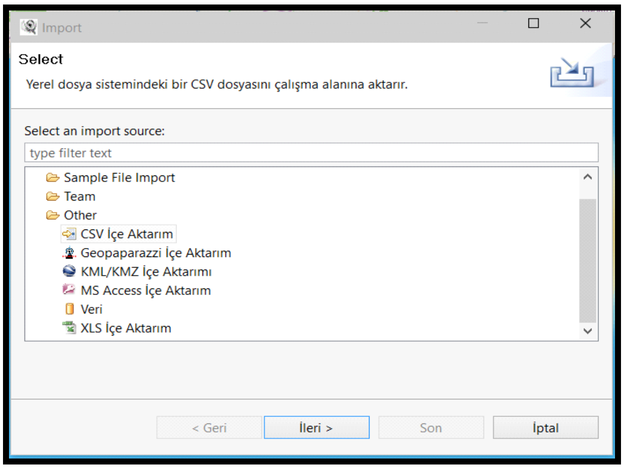


Figure H 37. Excel, Csv and Mdb Input Interface

In Figure H 38, users just only select the X and Y of data and then import the data in GIS software as a shape file. Other attributes can be defined as integer, string, text, double, float or one can leave them as default which is string.

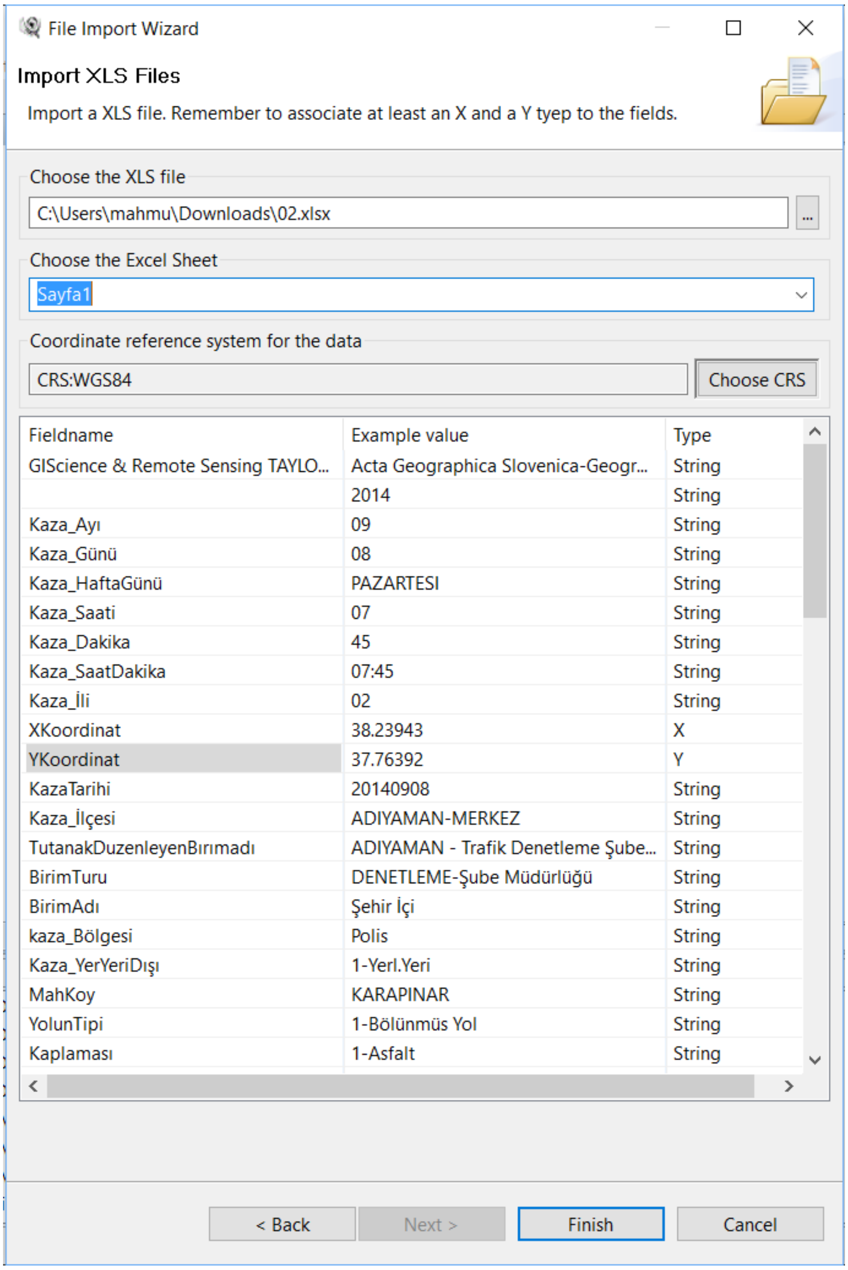


Figure H 38. X and Y Selection Interface

APPENDIX I

SCREENSHOT FROM THE OFICIAL WEB-SITE

Figure I 1 shows the main page of sda4udig study (www.sda4udig.com). The page includes the "Home", "About Authors", "Source Code", "Downloads" and "Form" pages. Authors page, gives the detailed information about the authors who are Prof. Dr. H. Şebnem Düzgün and Mahmut Çavur. One can download source code of RCaller, sda4uDig Plug-in and RScripts undr the Source Code Menu. Download menu includes the 32 and 64 bit Windows OS sda4uDig set up file. Finally, Form menu includes detailed document about study, user videos, surveys and useful links related with the study.

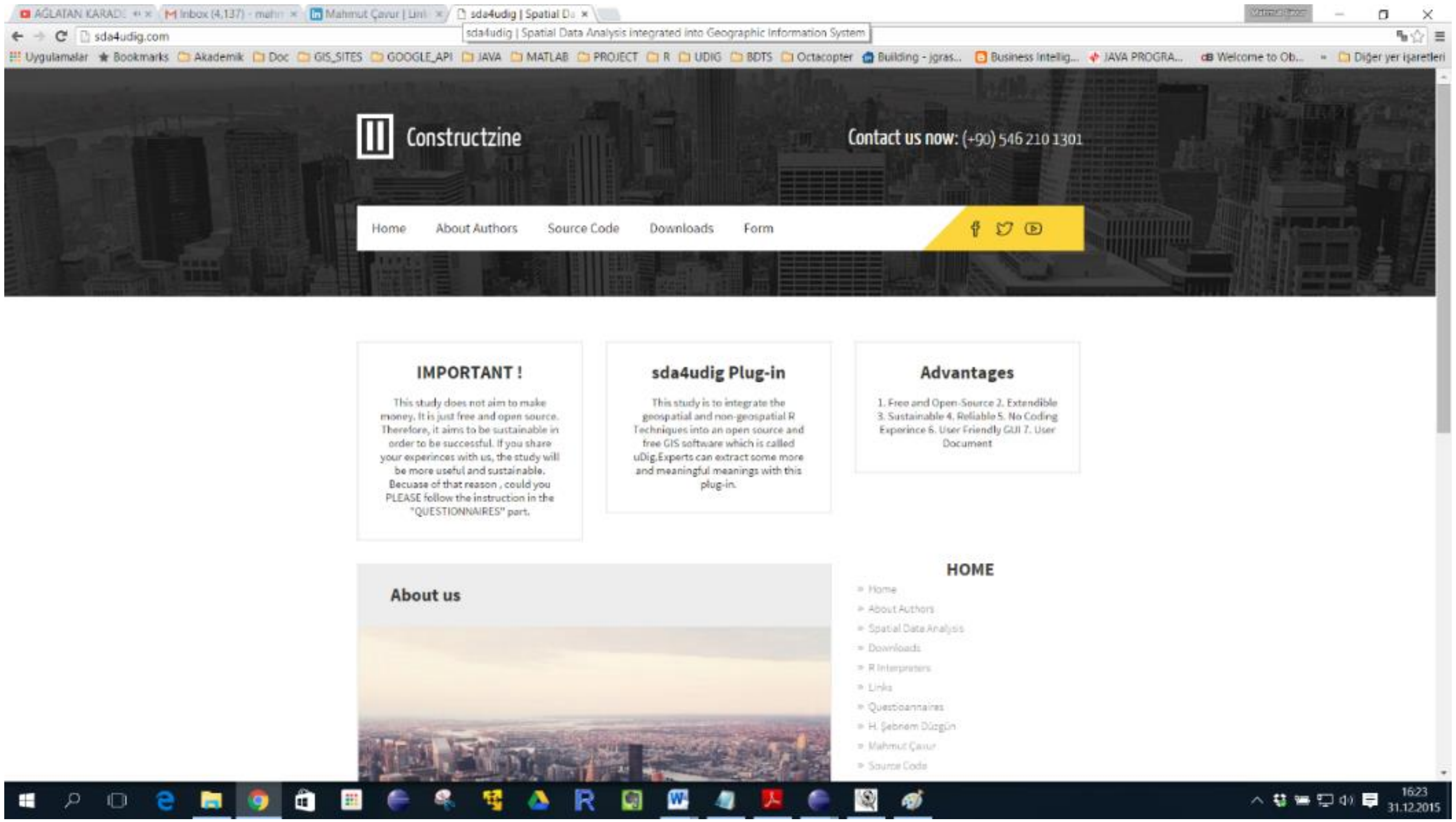


Figure I 1. Web-Site of sda4udig study

CURRICULUM VITAE

PERSONAL INFORMATION

Surname, Name:	Çavur, Mahmut
Nationality:	Turkish
Date and Place of Birth:	04.03.1984, Ankara
E-mail:	mahmutcavur@gmail.com

EDUCATION

PhD Middle East Technical University, Goedetic and Geographic Information Technology, 2016

M.Sc. Boğaziçi University, Management Information System, 2011

B.Sc Middle East Technical University, Mining Engineering, 2009

WORK EXPERIENCE

2011-2011	ESRI Turkey,	Project Engineer
2012-2012	Boğaziçi University	Thesis Study (BAP Project)
2012-2016	Kuzgun Informatics.	Managing Partner
2012-2016	MIneSoft Informatics.	Managing Partner

FOREIGN LANGUAGES

Advanced English

Mid-level French

PUBLICATIONS

Cavur, M.; Kemec, S.; Nabdel, L.; Sebnem Duzgun, H., "An evaluation of land use land cover (LULC) classification for urban applications with Quickbird and WorldView2 data," Urban Remote Sensing Event (JURSE), 2015 Joint , vol., no., pp.1,4, March 30 2015-April 1 2015. doi: 10.1109/JURSE.2015.7120486

Cavur, M., Özturan, M., Karaduman, C., & Içli, A. B. (2015). “Diffusion of GIS At Municipalities In Istanbul”. In IMCET, April 14-17 2015.

Chrysoulakis, N., Esch, T., Parlow, E., Duzgund, S. H., Tal, A., Sazova, A., Feigenwinter, C., Triantakonstantis, D., Marconcini, M. and Cavur, M., 2013. “The role of EO in sustainable urban planning and management: the GEOURBAN approach”. In Proceeding of the RSCy2013: First International Conference on Remote Sensing and Geoinformation of Environment, held in Pafos, Cyprus, April 8 - 10.

Cavur, M. and Duzgun, H. (2015). Geospatial and NonGeospatial Analysis of Trafic Accident Data A Case Study_EGM. In: 6. Karayolu TrafikGüvenliği Sempozyumu ve Sergisi. Ankara: Ankara Trafik Emniyet Genel Müdürlüğü, p.92

Cavur, M., & Duzgun, H.S. (Under Review). “A Framework to Embed Spatial Statistics Toolbox (R Techniques) in Open-Source GIS Software (uDig) Kernel Density Estimation Case Study”. International Journal of Spatial Science.