LSI-STAT - a Visualization and Analytics Platform

Thesis submitted in partial fulfillment
of the requirements for the degree of

Master of Science (by Research)
in
Computer Science and Engineering

by

Neha Pande
201202023
neha.pande@research.iiit.ac.in

International Institute of Information Technology
(Deemed to be University)
Hyderabad - 500 032, INDIA
July 2018

International Institute of Information Technology
Hyderabad, India

CERTIFICATE

It is certified that the work contained in this thesis, titled "LSI-STAT - a Visualization and Analytics Platform " by Neha Pande, has been carried out under my supervision and is not submitted elsewhere for a degree.

26/7/18
Date

Adviser: Prof. Dr. K S Rajan

To my Mother
for being my inspiration, my pillar of strength
and always encouraging me to do my best.

Acknowledgments

I would like to express my sincere gratitude to my advisor Dr. K S Rajan for being constant source of knowledge, and for his mentorship, support and guidance throughout the research. Constant discussions with him helped me gain new perspectives, explore new directions and think creatively. His constant encouragement has been a motivation for me throughout the journey which taught me a great deal about patience, hard-work and dedication.

I would like to thank my mother, father and brother for their continuous encouragement. To my Mom, this would not have been possible without you. Thank you for all the love and care throughout the entire period!

To my friends Ashwini, Jyoti, Sonali and Rashi for listening to me patiently, offering advice and being there with me all through the journey. Special thanks to Malini for her valuable inputs through all the discussions. I would like to thank all my fellow labmates and seniors who helped me and inspired me - Nishith, Sarthak, Gaurav, Manikanta, Tarun, Darpan, Mukul Priya and my friends Sakshee and Vartika.

I would also like to thank Dr. Suresh Munuswamy, Health Informatics Program Coordinator, IIPH-Hyd for making health insurance data accessible for the research.

I am grateful to my manager at Samsung R&D Institute Bangalore Mr. Siba for providing me a supportive work environment to be able to complete my thesis along with my job. To my colleagues at Samsung - Kapil, Aayushi, Mini, Prajakta, Saiksha and Priyanka for helping me.

To my advisor, parents, sibling, friends and colleagues, a big thank you to all!

The struggle you are in today is developing the strength you need for tomorrow, Don't give up.

Stay positive, work hard and make it happen!

Abstract

Geovisualization or geographic visualization makes patterns become apparent which otherwise would not have been deciphered. It prompts users into thinking and developing new hypothesis which leads them to make better decisions based on the visualization. It can be 2D or 3D visualization generated using real or simulated data. Moreover, both static as well as dynamic processes can be visualized using geovisualization softwares and tools. The visualizations are graphical representations of properties and relationships between objects in space and time.

Today, a large variety of geovisualization tools are available. Each of these have a different set of features. Study of current geovisualization tools was done to identify set of unique features. Based on this study, features of these tools were divided into data modelling, data analysis and data visualization. Data visualization helps us explore data, build and validate hypothesis. To gather further insights and identify patterns in this data, analytical functions are required. In some cases, you will have to model the data. Modelling is required to convert input data and develop a more sensible visualization. Different combinations of these features exist in various tools. Along with this, it is also important to understand other aspects such as whether the tool is map-based or graph-based, whether is open source or proprietary, what kind of data input format they have, whether it is interactive or not among others.

This work involves the design and development of LSI-STAT, a web-based spatio-temporal interactive analytical platform. It is a visualization platform which captures both spatial and temporal trends in map along with neighborhood. It is a SOLAP platform which extends OLAP capabilities to Spatial. It provides user with some basic querying capabilities based on OLAP principle. The platform models on user-given data. It computes data aggregation over field names based on hierarchy. This hierarchy is user-defined. It enables users to do an in-depth visual analysis of data to discover hidden insights and patterns in data. With this platform, it is expected that users can visualize charts in combination with maps spatially distributed over the area of interest thus providing for a more enhanced integrated spatio-temporal experience. It allows user to configure various parameters and define hierarchy and time-intervals as required. A large number of visualizations can be generated by the user with the help of this platform using location, time and attribute combinations. The developed platform allows user to select multiple data attributes and generate chart for geovisualization. With the toggle basemap option, user can change the basemap used in geovisualization. As geo-spatial extent of the data can change the perspective of looking at the data at multiple geo-levels, the information is tagged to these zoom levels. Also, we anticipate that as the usage of this tool builds, there can be some pre-defined functions and

some new toolsets may need to be incorporated. Hence, the platform has been entirely developed using a suite of open-source technologies.

To demonstrate the utility and functionalities of the developed tool and to provide for a good understanding of the value that it provides, two case studies have been presented in this thesis. To cover the range, 2 extremely different levels of case studies have been taken - local and global level. Additionally, both of them have different number of parameters. One of them is a GeoBI case study which was done on a large online retail data of three products sold in different countries during January 2009. The combined view of the temporal trends over the regions helps provide insights into 'what sells more where', 'is the growth patterns similar or not', etc. It is apparent that this approach helps overcome the fragmented view that the separate views of space and time using the earlier paradigm provided.

Another case study that was done using this tool was on Rajiv Aarogyasri health insurance data of Khammam district in Telangana for the years 2013 to 2015. Over the three year analysis which was done using this tool, it was seen that the map showed a lack of access to a large population. Results can help a policy maker to find places that are inaccessible, have limited access to healthcare facilities, lack certain healthcare services or have fewer number of healthcare facilities. Further, they can find the geographical extents of the population that are actually utilizing different healthcare facilities. By analyzing the geographical coverage and spatial distribution of existing healthcare facilities, they can take decisions for scaling up the existing healthcare facility network. It helps them to easily locate places to setup new healthcare facilities and identify services that need to be added to the healthcare facility. Also, capturing the temporal trends here, can be informative in understanding how these services or facilities fare or provide the right level of service.

The platform can be extended later to include more features such as time slider, word cloud, supporting multiple data input types, supporting overlay with external datasets such as road network etc. Interaction time can be improved by using tile-based rendering. Spatial predicate, neighborhood queries and displaying multiple graphs in a single chart can be supported in future.

Contents

List of Figures

List of Tables

Listings

Chapter 1

Introduction

1.1 Motivation

One of the primary and very important steps of data analysis is data exploration. It involves understanding of the overall data characteristics, the type of hypothesis it is likely to support, identification of interesting variables in the dataset and find variables that are correlated. An initial exploration can help in gaining an overall understanding of the data and its context. Based on these questions, different hypothesis can be generated and verified. Several interesting patterns can be discovered which were not obvious before. To explore data simply and quickly, data visualization can be used. This would enable them to easily recognize relations between different attributes, identify patterns and draw conclusions. According to Friedman (2008) the "main goal of data visualization is to communicate information clearly and effectively through graphical means. It doesn't mean that data visualization needs to look boring to be functional or extremely sophisticated to look beautiful. To convey ideas effectively, both aesthetic form and functionality need to go hand in hand, providing insights into a rather sparse and complex data set by communicating its key-aspects in a more intuitive way according to Friedman (2008). Interaction is an essential part of data visualization . With an interactive system, users can explore this data using query-driven approach. Highlighting details on demand, showing colorful web feature lines on moving mouse, rich pop-up windows with detailed information on user click and zooming in/out in the browser window on mouse click are some of the interactive features present in data visualization softwares and tools. Today, a large number of online tools and services are available for data visualization and exploration. However, the user-interaction in most of them is limited to panning or zooming to different levels in the map without any change in values displayed.

Traditionally, choropleth mapping techniques have been used which involve visualizing values over a geographical area in relation to only a single variable. They are based on statistical data aggregated over previously defined regions. A choropleth map consists of areas shaded according to the value of the variable. Choros in choropleth means space and pleth means value. Classification can be done based on equal intervals, equal frequency, geometric progressions, standard deviation or natural breaks. Several different types of progressions are used by cartographers for representing these [1]. This can be

blending from one colour to another, a single hue progression, transparent to opaque, light to dark or an entire colour spectrum. One another issue is that with the use of color is that one cant accurately read or compare values from the map [2]. These problems can be solved by representing multiple variables over a geographical region with charts. They can be used to visualize relationships and patterns between two or more variables.

With advancements in technology, a large amount of data is being generated such as meteorological data, geographical data, medicine, and data from different kinds of sensors placed in buildings, roads, cars, cell phones, and so on. Location-aware data is also easily collected using smartphones and other handheld devices due to the presence of global positioning systems (GPS). Out of this, approximately 80% of digitally generated data contains a spatial element [3]. Many of them include a timestamp. With increasing amount of data, there is a need for a tool which helps the user to quickly analyze such vast amount of information and generate hypothesis for both spatial and spatio-temporal queries.

While maps provide spatial visualization support, the time is only captured as a timeline-snapshot; while charts do capture the time based parametric variability, they miss out on the spatial interactions that these trends may also indicate. In addition, a static-linking between these two approaches, though possible, doesnt provide any clues regarding the patterns that might emerge at different spatio-temporal scales. As the boundary changes, the temporal data represented on the chart in the map changes.

Spatio-temporal data contains both spatial and temporal dimensions. Both these dimensions are composed of a number of attributes. These attributes are related in a particular order known as hierarchy. For example, day, week, month, year is a hierarchy. In this case, year is the highest level of hierarchy and day is the lowest level. Data aggregation for any dimension is done based on its hierarchy order. The values of the attributes lower in hierarchy are aggregated to an attribute higher in the order. For answering different types of queries, users need to view data at different levels of hierarchy. At a higher aggregation level, user can get an overall idea of the data whereas at lower aggregation level, helps user in answering specific queries related to the data. Drill-down analysis involves moving continuously from a higher to a lower level aggregation. On the other hand, roll-up analysis involve moving in the opposite direction from lower aggregation level to a higher aggregation level. For example, for answering queries- sales by world, then sales by country and then sales by city, where sales is an attribute involves performing a drill-down analysis. The value of sales by country can be computed by aggregating sales of all the cities in that country. In aggregation at a certain level, graph may show an increasing trend. However, on dis-aggregating them, they all can have different trends. Similarly, graphs in different spatial regions may show similar trends at a particular level, however on aggregation them to a higher level; the pattern may be lost. In some kind of aggregations, we lose out the pattern. Hence, it is important to understand the reasoning behind why the aggregations are similar or different.

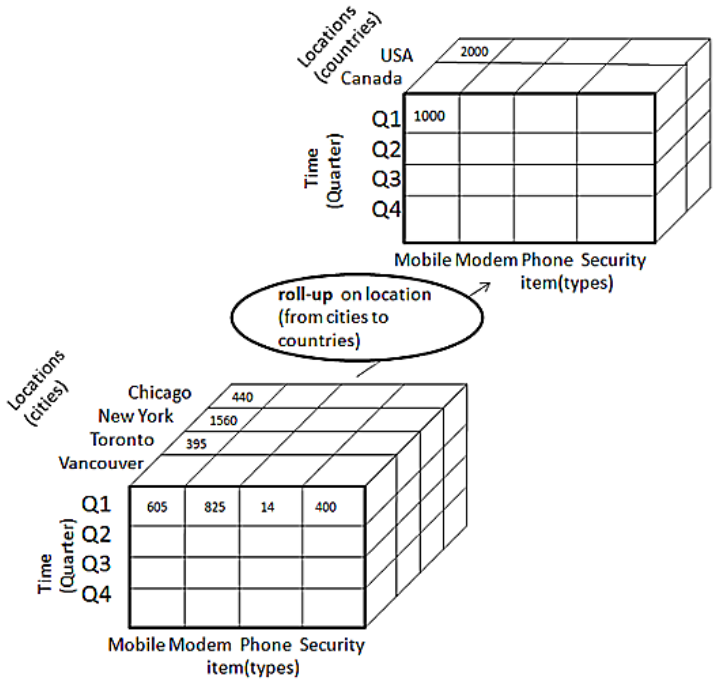

Figure 1.1: OLAP technology

OLAP (Online Analytical Processing) techniques help users to analyze data to perform visual exploration, see multi-dimensional view of data and perform drill up/drill down operations. An OLAP tool allows data to be analyzed from multiple perspectives and aggregation levels [4]. While OLAP technology is very useful for efficient multi-dimensional analysis, visual exploration and ad-hoc analysis, they still dont take geospatial characteristics for decision making. They visualize data values only with the help of pivot tables or graphs. They dont have a map component. Understanding spatial relations becomes very cumbersome because geographical factors are not considered [5]. Figure 1.1 shows how data is stored in an OLAP cube [6]. For example, if a user wants to know the regions where Modem was sold in the highest quantity. We can see that such spatial queries cannot be answered. Map component is necessary to find regions. The missing geospatial characteristics can be captured with the help of GIS. However, GIS on its own is not suitable for answering different queries across multiple dimensions. For example, Find districts where number of sales increased from the year 2012 to the year 2015? Some of the crucial operations such as temporal and spatial aggregation are not well supported in traditional GIS [7].

On combining GIS with OLAP tool features, Spatial Online Analytical Processing aka SOLAP systems can be formed [8, 9]. With SOLAP, data exploration and spatio-temporal analysis can be done easily. It enables data visualization by representing information on a map as well as in a tabular form. Representing information on map allows user to see context of data along with value. With interactive mapping, data values present on map are recomputed for changing geographic levels of hierarchy. User can explore data by navigating across the spatial dimension easily.

The problem associated with SOLAP is that it cannot materialize all possible geometric aggregations of spatial measures which in turn leads to storage space problems. Algorithms need to be designed to select spatial aggregation that needs to be materialized. Moreover, hierarchy needs to be drawn for spatial dimension. One of the more important characteristics is to look at topological relationships. This tells us the interactions between spatial objects.

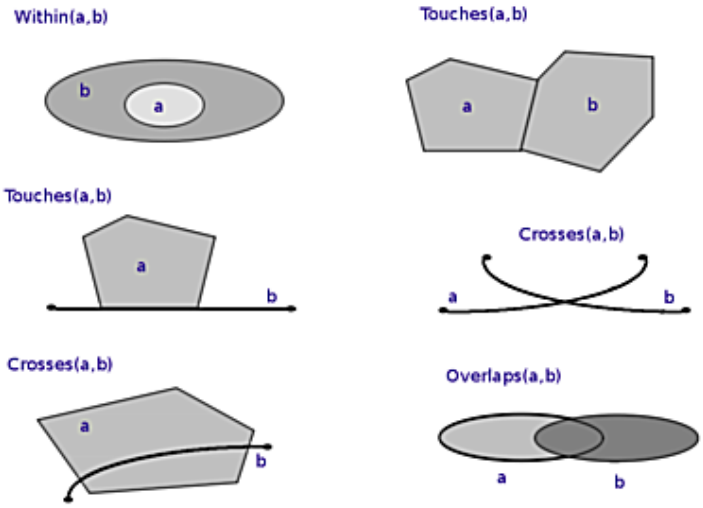

Figure 1.2: Topological Relations

Topological relations are relations of all objects which achieve the required conditions regarding spatial relations between objects. Figure 1.2 illustrates the various topological relations that are present [10]. For example, Select all buildings which lie in a particular region. Spatial relations tells us how a particular object is located with reference to another object. Examples for spatial relations are Equals, Disjoint, Intersects, Touches, Contains, Covers, Covered By and Within. Handling topological relations (i.e. overlap, inclusion) between members of same or different spatial levels remains a challenge. In addition to this, there are visualization issues like drill up/drill-down on either pivot table or graphical displays should reflect on one another. The most important feature of any SOLAP tool is visualization and also the main challenge [8].

Attempts have been done to capture spatio-temporal variation and integrate them to obtain different insights into the data. There are challenges of different kinds and in this thesis we are trying to address some of them. Some other characteristics like linking a map to a chart are also being attempted to be incorporated. In many cases, map and chart are often loosely linked and are separately processed. Some of these issues required to be further explored. Also, with changes in technology, they are to be adapted to suit such issues.

1.2 Research Questions

This thesis looks at some of the questions and would like to find solutions wherever appropriate. These questions are as follows:

- How do we incorporate spatial area aggregations aka. geo-hierarchy in the analysis? How do we create a linking between attribute data and geographical data? How do we compute data aggregation on field names based on hierarchy? Can there be a platform looking across time-scales and spatial-hierarchical levels?
- Can the maps be made more intuitive to the information they are presenting? How do we provide for a platform which allows user to choose from different visualization options? Can fields be added to the menu on the fly instead of precomputing them? How can the geo-related information change with changing geographic extents?
- Can technology be leveraged for such kind of intuitive visualization? Can such a platform be developed entirely using open source tools and technologies? Will this visualization help us gain new and interesting insights? With the help of such insights, can we identify any patterns that emerge?

1.3 Research Objectives

Towards developing a solution that can provide appropriate answers to some of the above questions, this thesis sets forth the following objectives to be addressed.

- Develop a web-based spatio-temporal interactive analytical platform which generates dynamic data visualizations based on user-given data and captures both spatial and temporal variations in map.
- Find a way to incorporate user-defined hierarchy in the platform and compute data aggregation on field names based on hierarchy. Develop a platform for handling of both the field level information and its aggregation options associated with the respective geo-hierarchies on a user need basis.
- Develop a flexible structure where user has an option to choose charts in combination with maps distributed spatially over the area of interest.
- Develop and show use cases where such a spatio-temporal data visualization platform can help provide valuable insights into the data and knowledge extraction.

While this thesis develops and implements a framework and proof of concept of the analytical platform and its application case studies, it desists from evaluating the cause of some of these interactions or

outcomes - which is out of the scope of this thesis. It is hoped that the development of LSI-STAT, as presented here provides the reader with an understanding of its various modules, its design characteristics and its appropriateness in building a spatio-temporal analytical and visualization framework.

1.4 Organization of Thesis

The chapters in this thesis are as follows:

- Chapter 2 describes the background survey done of existing visualization tools.
- Chapter 3 details the design and development of this tool including the different approaches considered, technologies chosen and architectural overview.
- Chapter 4 explains the different modules of LSI-STAT and their working.
- Chapter 5 describes how the tool is useful for online retailers to explore their data, understand relations between different variables, build and test hypotheses through data visualization and analyze.
- Chapter 6 shows the application of this tool on a case study done on Rajiv Aayogyasri health insurance data for Khammam district in Telangana.
- Chapter 7 concludes the thesis by summarizing the work done in this thesis.

Chapter 2

Literature Survey

2.1 Geovisualization

Geovisualization is short for Geographic Visualization. It refers to set of tools and techniques supporting analysis of geospatial data through interactive visualization. It involves use of techniques for exploring data, forming hypothesis and developing solutions. With the help of visual exploration, we can make better decisions by display patterns along with interaction and dynamics.

Figure 2.1: Simple Openlayers Map

Geovisualization has roots in cartography. Scientific visualization, image analysis, information visualization, exploratory data analysis, urban simulation and GI Science are closely related to field of geovisualization. It is different from cartography and map production since it typically involves using interactive data exploration tools and softwares. Together with GIScience (Geographic Information Science), it is able leverage data resources to meet scientific and societal needs and develop visual methods and tools for geospatial data application.

Figure 2.1 shows a simple openlayers map [11]. Openlayers is a popular javascript based mapping library. Such a map is able to intuitively give us some interactions. However, it still is only a single timeshot and the dimensionality of the map limits the kinds of things you see on it before you get into methods.

Visualization term was first used in an article by geographer Allen K. Philbrick of the University of Chicago in 1953. The term was redefined in 1987 by National Science Foundation and was placed at the

convergence of computer graphics, computer-aided design, signal processing and user interface studies. Geovisualization developed in the early 1980s as research field mainly because of the work of graphic theorist Jacques Bertin.

Today, maps are not static and the user can directly control various aspects of map display due to the rapid development of spatial technologies such as GIS, multimedia atlases and 3D virtual globes. The term geoviz has become a significant area for applied research by GIScientists and of relevance to human geographers more generally in how they engage on spatial media (Dykes et al 2010; Thielmann 2010). Geovisualization is the most important development in cartography since the thematic mapping revolution of the early nineteenth century (MacEachren 1995: 460).

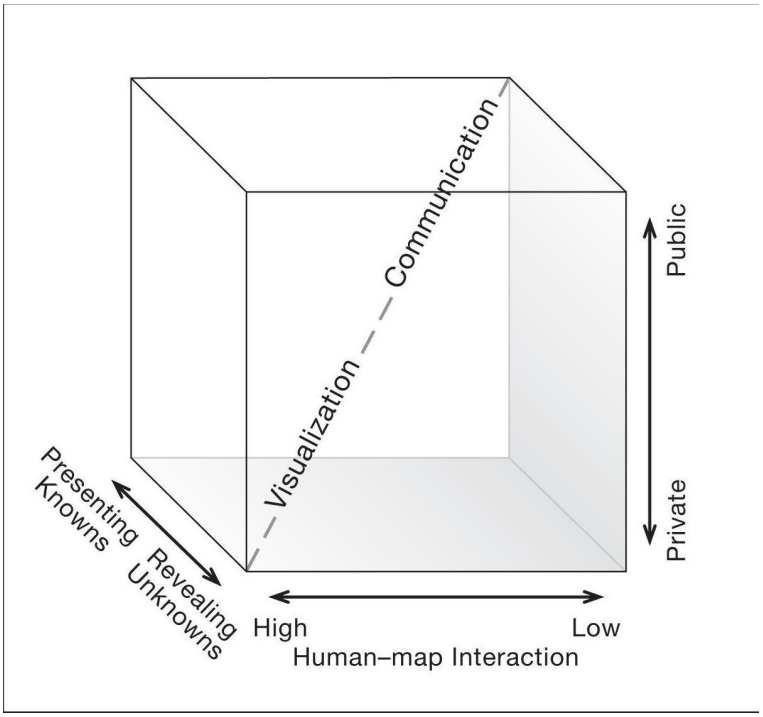

Figure 2.2: Cartography Cube
[12], Visualization in modern Cartography, Pearson Prentice Hall, 2009

MacEachern (1984) provides a cartography cube with three axes to encapsulate the distinctive characteristics of visualization. The first axis covers the scope of the user audience for the geovisualization, next is the degree to which the map offers interaction in use and third axis is data relations. Distinct types of geovisualization practices can be classified and placed in cartography cube - explore, analyse, synthesize and present. Figure 2.2 shows a cartography cube. They are present from the lower corner of the cube to the other corner (MacEachren 1994: 3, original emphases). The explore helps us answering questions like - What is happening here?. They are highly interactive maps which prompt users to think, generate hypothesis and validate them. The analysis phase uses interactive maps to process and classify complex data and break it down to reveal unknown patterns. The synthesis stage requires evidence to be assembled for supporting certain hypothesis.

Static maps have been used traditionally for geovisualization. However, they have limited exploratory capability. Geovisualization along with GIS helps in generating more interactive maps with which user can explore different layers of the map, zoom in or out and change visual appearance of the map.

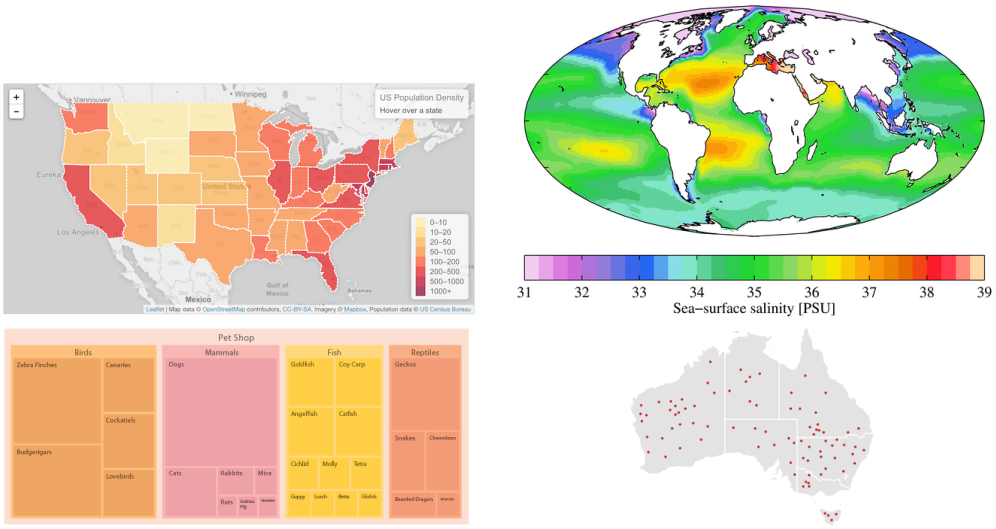

Figure 2.3: Types of Maps

One of the oldest techniques is the choropleth mapping technique which involves representing data using different colors or shading patterns [13]. Every color or shading pattern represents a particular value or range of values. Figure 2.3 shows different types of maps- choropleth map (in top-left corner)[14], heat map (in top-right corner)[15], tree map (in bottom-left corner)[16] and dot map (in bottom-right corner)[17]. Here, the user has to choose the number of classes and the data classification method. This classification method can be equal interval, equal count, natural breaks, standard deviation etc [13]. Chloropleth maps have a bias towards larger areas. They assign larger visual weights to such areas (Dent, 1999; Dorling, 1996, Speckmann/Verbeek, 2010). Another technique involves using heat maps for representing large datasets consisting of continuous data on map [18,19].

Geographical boundaries are not visible in heat maps and a color spectrum is used for representing the data [20]. It is very useful in identifying patterns and hotspots. For example, they can be used to represent amount of average rainfall in a region. Other types of maps which are frequently used are hexagonal binning, dot maps, tree maps, cluster maps, radius maps, bubble maps, cartogram maps etc.

Some of the data available today is largely unstructured. For such data, database such as NoSQL is being used to store this [21]. Also, some of them have clear definition of geometries whereas others dont. This data is very noisy [22]. There is a need to clean this data first and then analyze it. Integration of this data with other structured data also remains a challenge [23].

Different models of geovisualization, analytics and SOLAP based approaches are being tried by different research groups [24, 25, 26]. It is not just data visualization but the type of questions that

they are looking at. In some kinds of visualizations, topological relations are often used such as trees, flow charts, network diagrams, path diagrams, path diagrams etc but not in others [27]. These are either explicitly represented using arrows or shaded areas or are implicitly represented [27]. Topological relationships are the focus of much of the research [28, 29]. One issue is of measuring aggregation and analyzing topological relationships between hierarchy levels [29, 30].

Merging BI and GIS helps in exploratory decision making by enabling exploration of spatial relations between data and observing and interpreting phenomena by representing them on map. This led to development of geo-analytical tools with an ETL Tool such as GeoKettle [31]. ETL stands for Extract, Transform and Load. Extract refers to extracting data from one or more data sources, transform refers to correcting errors, cleansing data, changing data structure, making data compliant to defined standards etc and load refers to loading transformed data into the target database. ETL tools are responsible for insertion and updation of data. GeoBI stack is Pentaho open source BI software stack developed by GeoSOA group and which consists of Spatial ETL system (GeoKettle), SOLAP tool (GeoMondrian), Spatial Data Mining (Weka), Data Warehouse (PostGIS and Oracle Spatial), Reporting tools (Pentaho Reporting) and SOLAPLayers (opensource project) [32]. GeoKettle is "spatial-enabled" version of Pentaho Data Integration (Kettle) [31]. It is part of geospatial BI software stack which includes GeoKettle, GeoMondrian and SOLAPLayers [31]. It is an open source powerful, metadata-driven spatial ETL tool and GeoMondrian is an open source spatial online analytical processing server and is a spatially-enabled version of Pentaho Analysis Services (Mondrian) [31, 33]. It is an implementation of SOLAP server[33]. It provides a consistent integration of spatial objects into the OLAP data cube structure instead of fetching from external spatial DBMS, web service or a GIS file [33].

2.2 Examples of existing toolkits

In this section, some of the existing tools which look at maps and various kinds of geovisualizations are being presented for our understanding of the kind of features or functionalities that they have.

2.2.1 Tableau

Figure 2.4: Tableau User Interface

Tableau is an interactive visualization focused on business intelligence [34]. It has a large number of features. It allows users to connect to a large number of servers such as Tableau Server, Microsoft SQL etc. [35] Users can import data from excel, text file, access files, and many others [35]. Table and map can be viewed side-by-side. It supports a number of operations such as cross-join, inner, left, right and full outer joins. Also, user can change datatype of a field or split a field very easily. Rows and columns can be interchanged simply. It supports multiple aggregation functions such as sum, max, min etc. The attribute hierarchy can be set by the user. Quick aggregation and disaggregation is possible on the fly [36]. Some of the quick calculations like year over year growth are supported. It is possible to connect to another table and perform any join operation. These attributes in both tables can be renamed. Table fields can be easily split by using any separator and keep the required part of the field. Multiple graphs can be added in a single figure. Colors can be chosen from the palette available. A wide variety of options are available for color, label, size, detail and tooltip. Tableau provides users to create advanced chart types such as waterfall, bump, funnel, pareto charts etc. Categories in data can be sorted. They can even be grouped for visualization. User can set his own attribute hierarchy which helps in drill-up and drill-down. Zoom by Area feature is present to see data points in detail. It is possible to publish workbook with Tableau Server or Tableau Online.

2.2.2 GeoVista CrimeViz

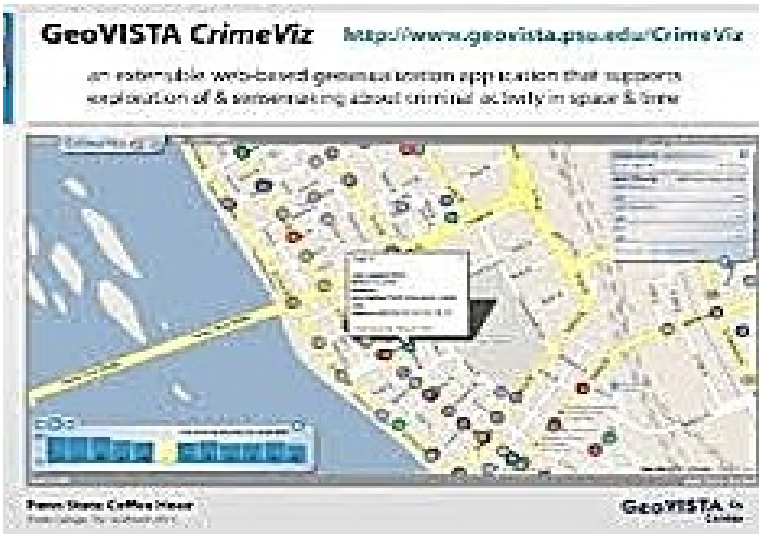

Figure 2.5: GeoVista CrimeViz User Interface

GeoVista CrimeViz is an extensible web-based map application that supports exploration and sense-making about criminal activity in space and time [37]. It consists of a central interactive map. The basemap used is a Google Map [37]. It supports interactive filtering by crime type and has linear and composite animations (temporal). It comprises of a set of togglable map layers. Three panels are present - map, data layers and temporal panel. On point click on the map, all information about the datapoint is displayed. Vector layers of different geometry types such as point, line and polygon can be overlayed. The temporal unit for animation can be changed. For example- week, year. The histogram corresponding to the time animation is shown on right side. Multiple point layers can be shown on map. Clicking on an area of histogram shows data values of that point. The map has panning and zooming controls. Reset panning option is also present. Single-click on the crime incident opens an information window containing details about the case record. A double-click zooms the map to the nearby region. The user can then open Google Street View to view the location near the crime. A drop-down list on the left of the temporal controls panel is used to select the temporal aggregation unit for the display (week, month, or year) [37]. A map with linear time binned by months; the time graph shows one bar per month across a span of about four years. Color represents the number of crimes for each crime type in a given composite day. ActionScript code packaged as zip present on website which can allows user to write their own webapp.

2.2.3 LANDIS-II User Interface

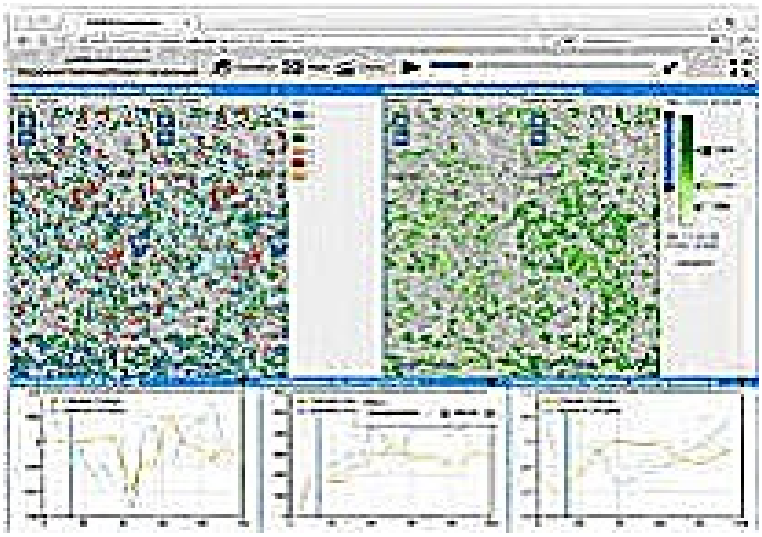

Figure 2.6: LANDIS-II User Interface

LANDIS is a forest landscape model which simulates forest growth, competition, seed dispersal succession and disturbances including fire, wind, harvesting, insects, global change across landscapes [38]. It represents them as grid of cells and tracks age cohorts of each species rather than individual trees [38]. It simulates distinct ecological processes, allowing complex interactions to play out as emergent properties of the simulation [38]. LANDIS-II is a modeling environment in which users plug in extensions that independently simulate specific processes or output specific cell or landscape attributes, each operating at a user-defined time step that is consistent with the process being modeled [40, 41]. Some of the types of succession extensions include age-only, biomass, forest carbon, bFOLDS, pNet and Net Ecosystem CN [40, 41]. Disturbance extensions available are fire, wind, biological disturbance agents, harvest, drought etc [40, 41].

It has preloaded data and doesnt allow the user to upload data [41]. For data visualization and analysis, it has 3 options - scenarios, maps and charts [41]. User can select multiple scenarios for visualization. Each scenario is associated with a map. Example- paper production, business. Legend is displayed for all maps. Color shades for different classes can be seen along with values (starting and ending) for those classes. More than one type of charts to be displayed can be selected to be displayed. The charts are displayed below the maps separately and not one the maps. The chart type used is line. Class interval values are adjustable and modifiable. Color-coding scheme can be adjusted for each of the individual set of data points. While playing timeline, user can simultaneously zoom-in/out and pan into map.

2.2.4 EWGAT

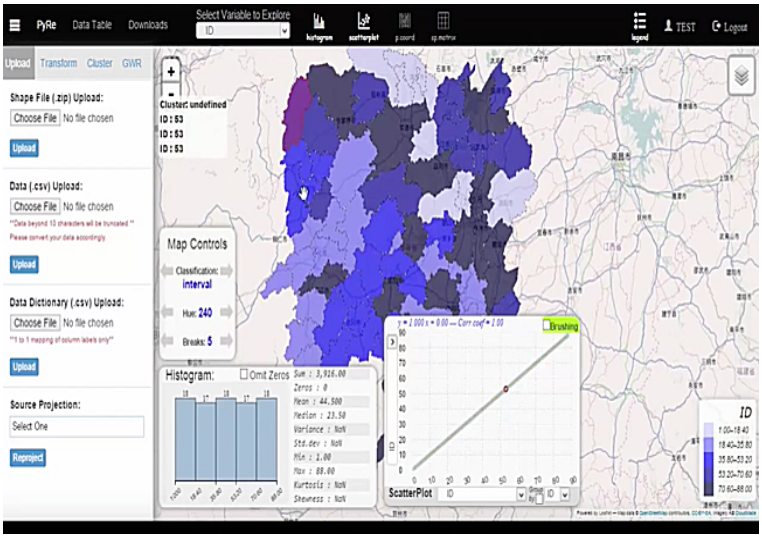

Figure 2.7: EWGAT User Interface

EWGAT stands for extensible web-based geospatial analytics toolkit. It can be used to calibrate, present and synthesize Geographically Weighted Regression (GWR) [42]. Technologies involved in its development are D3.js, jQuery, Apache Tomcat, R, JSON, GDAL, xls, jsp, html, css, leaflet, js, backbone.js, bootstrap, json and python [42].

The tool accepts shapefile in zip format and data points in csv format [43]. It supports only choropleth mapping. It allows user to select this join variable. Map controls given to the user are classification type and hue. It displays a histogram and a scatterplot by default. However, they can be changed. On selecting parts of histogram and scatter plots, the respective areas in choropleth map get selected. Global and local geographic regression values can be computed. User can select variables for each of these. Other customizable options are choice of kernel function, bandwidth, selection criteria, distance type and variables (both dependent and independent). The result is shown as 2 choropleth maps side-by-side. On clicking parallel coordinates option, graph along with table is shown. On clicking report option, commonly computed statistics for all variables are shown such as min, mean, median, 1st quartile, etc. Area of shapefile on map will get highlighted on mouse click. Similarly, table values get highlighted on moving mouse over certain region of graph. In addition to this, user can view datatable and reproject the file.

2.2.5 TransAtlas

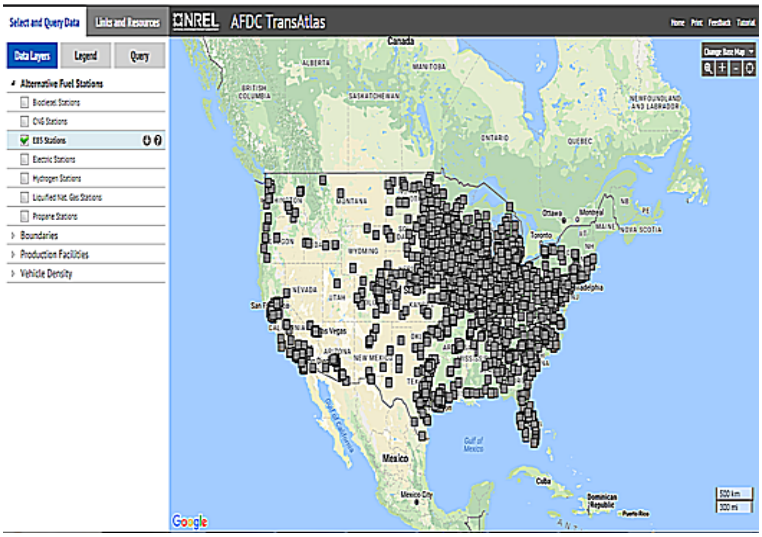

Figure 2.8: TransAtlas User Interface

In TransAtlas, the input data is pre-populated data [44]. Input data is available for download in more than one geospatial data formats including CSV, Shapefile, KML, GeoJSON along with source of this data [44]. It supports multiple data layers. Data layers can be overlayed. All layers have point data. Boundaries present as polygon layers. They include counties, state borders, congressional districts and clean cities coalitions and are available for download along with source of data. Clicking on the data point shows information about the data point. User can select and query data. Queries supported are point, region, custom shape and attribute. Predefined color coding scheme is present. However, it can be modifiable by user. Zoom operations (zooming-in and zooming-out) are supported. With the help of the zoom by area feature present, user can zoom into any area of the map. The base maps supported are none, Openstreetmap, Google Hybrid, Google Terrain, Google Satellite, Google Roadmap, CARTO Dark, CARTO Light. User can find a location by lat/long or address, city, state, zip. Extra links and resources are also listed.

2.3 Analysis of existing toolkits

A large variety of geovisualization tools are available today. Each of these tools have a different set of features. A study of these tools and their features was done. Based on this study, it was found that these features could be broadly categorized into data visualization, data analysis and data modelling based. Data visualization helps us explore data, build and validate hypothesis. To gather further insights and identify patterns in this data, analytical functions are required. In some cases, you will have to model the data. Modelling is required to convert input data and develop a more sensible visualization.

Also, with the help of this study, popular and unique geovisualization tools were identified. They were TransAtlas, LANDIS-II, Tableau, EWGAT and GeoVista CrimeViz. Each of these tools represents a different category of tools.

Some of the important and unique data visualization, data analysis and data modelling features have been compared in the tables below. Tables 2.1 and 2.2 give a comparison between data visualization based features. Similarly, tables 2.3 and tables 2.4 and 2.5 show the comparison between analysis and modelling based features respectively. The presence and absence of features is indicated by Y and N in the tables respectively.

2.3.1 Data Visualization

Toolkit	UI design	Legend	Charts on map for each data point	Timeslider	Zoom
TransAtlas	2D	Y	N	N	Y
LANDIS-II	2D	Y	N	Y	Y
Tableau	2D	Y	Y	Y	Y
EWGAT	2D	Y	N	N	Y
GeoVista CrimeViz	2D	Y	N	Y	Y

Table 2.1: Data Visualization Features (1)

Toolkit	World Cloud	Heat Map	Radius Map	Dimensionality Reduction
TransAtlas	N	N	N	N
LANDIS-II	N	N	N	N
Tableau	Y	Y	Y	Y
EWGAT	N	N	N	N
GeoVista CrimeViz	N	N	N	N

Table 2.2: Data Visualization Features (2)

From table 2.1 it can be seen that all these tools are 2D and legend is present in these tools. In most of them, color shades for different classes can be seen along with values (starting and ending) for those classes. They can be modified by user. In LANDIS-II, user can visualize data in two maps simultaneously and both of them having common legend. This makes it easier to understand and compare both maps. Charts are frequently used by many tools to show trends and patterns. However, many of them don't display charts with geolocation. Of the tools which were studied, only Tableau is able to display

charts on map for each datapoint. Out of these, LANDIS-II, GeoVista CrimeViz and EWGAT display charts alongside maps. In EWGAT the type of graph is fixed - histogram and scatterplot. In LANDIS-II charts are displayed below maps separately and the chart type is fixed to line. TimeSlider functionality is supported by LANDIS-II, Tableau and GeoVista CrimeViz but there are slight variantion in how it is used. In LANDIS-II, zoom and pan operations on map can be performed during time animation. CrimeViz allows for both linear and composite animations to be displayed. Histogram is also shown on the right side along with time animation. A drop-down list on the left of the temporal controls panel is used to select the temporal aggregation unit for display (week, month or year). Data is aggregated into bins based on the choice. Controls such as Play/Pause, Step Backwards and Step Forwards are present. In Tableau, time slider doesn't show changes in map continuously but only on mouse click. Zoom operations (zooming-in and zooming-out) is supported in all the tools.

From table 2.2, it can be seen that word cloud feature is available only in Tableau. Heat maps and radius maps are commonly used to represent data. Tableau supports different types of charts. Advanced chart types which are supported include waterfall, bump, funnel, pareto charts etc. Dimensionality Reduction feature has been recently added in Tableau. It can be done by using Principal Component Analysis (PCA) in Tableau by using TabPy. These features are not present in any of the other tools.

2.3.2 Data Analysis

Toolkit	Assoc. Rule Mining	Spatial Predicates	ML algo (inbuilt support)	Spatial Clustering	Predictive Analysis
TransAtlas	N	N	N	N	N
LANDIS-II	N	N	N	N	N
Tableau	N	N	Y	N	Y
EWGAT	N	N	Y	Y	N
GeoVista CrimeViz	N	N	N	N	N

Table 2.3: Data Analysis Features

Table 2.3 represents data analysis features. Features such as association rule mining and spatial predicates are not supported in any of the tools. Adding these features to the tools would further enhance the features. This feature is not supported directly in Tableau. However, if association rules are to be created, user can connect to a data mining tool such as Weka and then create static association rules. Spatial join is planned to be added in later releases of Tableau. Tableau has recently added support for machine algorithms in the tool. EWGAT allows user to perform geographic weighted regression using features present in the tool. Out of those surveyed, only EWGAT supports spatial clustering. Predictive Analysis feature is present only in Tableau.

2.3.3 Data Modelling

Toolkit	Overlay	Multiple map types	Multiple data input types	Re-project CRS
TransAtlas	Y	Y	N	N
LANDIS-II	N	N	N	N
Tableau	Y	N	Y	N
EWGAT	N	N	N	Y
GeoVista CrimeViz	Y	Y	N	N

Table 2.4: Data Modelling Features (1)

Toolkit	Query	Geocoding (in-built)	Connection to DB (internal / external)	Agg./Dis-agg. on zoom
TransAtlas	Y	NA	N	N
LANDIS-II	N	NA	NA	N
Tableau	Y	Y	Y	Y
EWGAT	N	Y	N	N
GeoVista CrimeViz	Y	Y	Y	Y

Table 2.5: Data Modelling Features (2)

Table 2.4 represents data modelling features such as overlay, multiple map types, multiple data input types and reproject CRS. Out of the tools, TransAtlas, Tableau and GeoVista CrimeViz allow users to overlay multiple data layers. However, not all of them allow overlay of multiple geometry types. For example, all layers have point data only in TransAtlas. In GeoVista CrimeViz, vector layers of different geometry types (point, line and polygon) can be overlayed. Different basemaps allow user to visualize data with different backgrounds to give a different insights into data. TransAtlas and GeoVista CrimeViz have this feature. TransAtlas supports basemaps - Openstreetmap, Google Hybrid, Google Terrain, Google Satellite, Google Roadmap, CARTO Dark and CARTO Light. This is not a visualization feature and has been listed as a modelling feature because the user data and the basemap may not be in the same projection system. So, there is a transformation or conversion that is required to be done at the backend. Hence, it is a modelling feature of the data. Some of tools allow users to upload input data in more than one format such as csv, txt, shapefile, kml, postgres and geojson. Out of the tools in this survey, only Tableau has this feature. Others such as EWGAT have fixed data input format. EWGAT

accepts shapefiles in zip format and data points in only csv format. In Tableau, there is support for users to connect to a large number of servers including Tableau Server, Microsoft SQL etc. Additionally, users can import data from excel, text file, access files among others. Reprojection refers to direct conversion of one map projection system to another. In Tableau, there is no direct conversion option. However, user can follow a series of steps manually to convert and adjust projection to tableau basemap.

Table 2.5 represents data modelling features such as query, geocoding, connection to database and aggregation/disaggregation on zoom. Of the selected tools, TransAtlas, Tableau and GeoVista CrimeViz allow user to query data on map. If user data contains address, there is a need to geocode data before the data point can be added to map. Tools EWGAT and GeoVista CrimeViz have inbuilt geocoding feature. In Tableau, geocoding is restricted to state/province level. Geocoding of street address is not done by the tool. Tools - Tableau and GeoVista CrimeViz allow connection to database. Connection to external database of any kind is not applicable for LANDIS-II. Aggregation and dis-aggregation on zoom is present only in Tableau and GeoVista CrimeViz tools.

2.4 Observations and Summary

Based on visualization, most of the tools can be classified into three categories. First category is of those tools which are only mapping and 2D data based. Example EWGAT and TransAtlas. TransAtlas uses choropleth maps (color-coded differences). Second category of tools have locational information integrated with temporal data but it is only for single point locations. Extending it to multi-point location is still a challenge and has not been done by the current tools. They dont have such capabilities. Therefore, it is difficult to understand spatial interaction, while temporal one may be understood. Most of these tools have temporal capabilities but rarely have spatial ones. Example LANDIS-II, GeoVista CrimeViz. GeoVista CrimeViz also shows aggregation and dis-aggregation with zoom. Thirdly, understanding chart with geolocation is different from understanding chart without geolocation. 2D maps provide charts that are static in nature. Hence, for visualizing another time, charts need to be regenerated and shown. These should be able to show spatio-temporal interactions along with their neighborhood simultaneously. Example Tableau.

There are also many other differences in the tools. Some of them such EWGAT and LANDIS-II are open source whereas others are not. Data is pre-stored in tools such as LANDIS-II and TransAtlas. Most of them are map-based and not graph based like EWGAT, TransAtlas, LANDIS-II and GeoVista CrimeViz. Even in graph-based tools, some of them will be able to represent multiple geometries at a time such as Tableau whereas others wont. It is important to capture neighboring trends and geometries. Not all tools are able to do so. The data input format is different for all tools. Tools such as EWGAT support only certain fixed formats whereas some others allow users more flexibility. Some tools are highly interactive whereas others are moderate or low. If preprocessing is involved, then they may save time and increase interactivity which is not the case. In many cases, the response of the analytical outcome is not quick enough.

It can be seen that not all features are present in most tools today. In any kind of geovisualization that is currently available, we can catch a couple of them and we lose some of them. Even a simple feature such as drill-up and drill-down is implemented differently in different tools. It could be geometry based hierarchy or attribute hierarchy. In most tools, the levels can be defined by using only attributes from input data and are generally based on fixed geometry. Thus, limiting the interactivity for end-user.

Chapter 3

Development Methodology of LSI-STAT

We developed a web-based spatio-temporal analytical tool (LSI-STAT) in order to provide for spatio-temporal analysis. The main goal is to see how space and time can be captured and visualized together. For this it introduces an interactive map with online analytical processing (OLAP) and computes data aggregation based on user-defined geo-hierarchy with dynamic visualizations in a defined framework. Additionally, an integration of the attribute based charts to be spatially linked to the geo-hierarchy of the map extent that is being displayed was done. An option to aggregate or analyze over single or multiple attributes was also added.

In the current set of visualization tools today, interaction is limited to only panning and zooming to different levels in the map with no change in values displayed. Also, most of the user data has various attributes collected over time across geographic detail. This data is hierarchical in both spatial and attributes. To do a visual analysis of this data, there needs to be a system which can aggregate over this data at various levels. The developed system attempts to find solutions to these problems. The main idea is to develop a spatio-temporal analytical tool and this chapter describes how to achieve that.

3.1 Overall System Design Criteria

For designing LSI-STAT, system usability, user-friendliness and performance were the main criterion which were used. Some of the existing tools require users to have knowledge of GIS to put together things and carry out the analysis. While designing this tool, it was kept in mind that the end-user may or may not have any knowledge of GIS and any user should be able to use this tool. Also, some of the tools may require user to install additional software or undergo training in order to learn how to use various functionalities of the tool. So, a browser-based system was developed which allows the user to visualize results on map. With this, a much wider audience can use this tool. Performance here refers to the overall system performance. The system should display result on map for any user-uploaded data and user-selected options.

3.2 Overall System Design Approach

This subsection explains the various approaches considered for designing this tool, their advantages and limitations and the reasoning behind the choice of this final approach. The final developed tool is able to fulfill all the objectives mentioned in section 3.3.

3.2.1 Server-side vs Client-side

For developing this system, the choice of developing a client-side system or a server-side system was to be made. In a client-side system, most of the processing is required to be done by the clients browser. In a server-based system, clients browser would send request to the server along with the required data, and the server will do the major portion of the processing. The server would reply back to the client with the processing results. Both types of systems have their own advantages and disadvantages.

Even though the basic data is coming from the user, the challenge is in the generation of the chart and the multi-level aggregation that has to be done. If these are to be done at the client-side, they would require a very large amount of computation has to be done. To reduce that and have a seamless way of visualizing the data, we have chosen the server-side. On the client-side it could add up to a heavy load on the clients browser. If the browser is unable to handle this load, the system may stop working altogether. It is not a question of data but a question of the number of operations that are to be done. If client model is to be chosen for implementation, client device characteristics need to be considered. This is because the technologies chosen, browser properties and the way it is implemented is not uniform across all platforms. In order to keep it platform independent, we chose the server model. The geo-hierarchical effects if computed on the client side, can lead to a more jagged (non-seamless) interaction for the user as the client may need to compute and recompute at every zoom level or may need a large cache to handle such data, which is less predictable at the design stage. Moreover, performing computation on the server-side is much faster than doing so on the client-side. Some of the tasks can be done in multi-threaded environment or as background tasks on the server but may not be possible to do seamlessly on the client. And since this is a browser based system, it would not require the clients system- hardware, browser, and plugins to have any minimum requirements or install any kind of software.

3.2.2 Interactivity

Data represented on the map changes as the user drills up or down into the map. So, there arises a need to generate new charts each time. These charts can be generated on the fly or they can be pre-generated and stored. The advantage of preprocessing these charts is that the tool interactivity is much better compared to generating them on the fly. With a CPU having higher processing power, charts can be generated quickly. These can be saved locally in a chart cache and then added on the map when required.

3.2.3 Choice of a file based system

There are two options available- storing data in files or in database. From the current design principle perspective, we chose the file-based system. The user can define the file as he wants. And he doesn't have to upload the schema everytime before uploading the data. The file itself is considered as the primary source of data. We give control to the user for defining the data rather than adopting a more schema based solution on how data is structured. Additionally, this approach avoids linking of files.

3.2.4 Chart based visualization

Large amount of data can be conveyed in an easy to understand format that communicate the relevant information to the audience. Variables represented on the chart can be absolute values, percentages, frequencies and categories. Depending on the type of data you are presenting, user can choose a bar, line or pie chart. In this tool, charts are being used to represent different attributes and temporal data and map for representing spatial data.

A bar chart is composed of discrete bars that represent different categories of data. The height of the bar represents the quantity within that category. They can be vertically or horizontally arranged and are useful in comparing values across categories. It can explain multiple attributes present at different time-periods. Along with this, we can also see inter-attribute comparison at the same time. One example is to use bar graphs to represent numbers of sales by different sales representatives. A line chart is useful in illustrating trends over time. Continuous data values can be represented easily. One example is to represent number of sales by a sales representative in different months of the year. A pie chart is circular and can be divided into sections to represent proportions. It can be used to visualize part of whole relationship.One example is to visualize how a person spends his earnings. With pie charts, we cannot see temporal data along with the attribute data. We will have to generate different pie charts to visualize temporal data at the same time. So, for visualizing temporal data along with the attribute data, only line and bar charts are suitable.

3.2.5 Choice of Technology

While choosing technologies for implementation, the following factors were considered - popularity, how old is the technology being used, support for different environments and integration with other technologies chosen. An initial analysis of popular technologies for implementing was done and suitable ones were shortlisted. The rationale behind this is that popular technologies have a larger user community which ensures that the technology doesnt become redundant in the long run and if there are any bugs found they would get fixed.

The developed prototype was hosted on Apache Tomcat 8. It is one of the popular hosting servers and supports web application deployment across for different environments. It is a Java based application server. To run a PHP application in this environment requires closer integration between Java and PHP. This was provided by PHP/Java Bridge which was used in Apache Tomcats hosting environment.

3.2.6 User-Interface Design

Given that we are planning to build a map and a chart which are 2 different features coming from 2 different servers, the interlinking of both of them has to be captured in the user interface design. Placement issue is an important design consideration. This can be done in one of three approaches - placing them side-by-side, up and down and one over the other. We chose the third approach because we are not only interested in a single point but also its locational information and neighborhood.

3.3 System Objectives

In order to provide a way to visualize temporal patterns, Lab for Spatial Informatics - Spatio-Temporal Analytical Toolkit aka LSI-STAT was developed with the following objectives:

Integrating locational and temporal patterns using principles of geo-hierarchy

- Incorporate spatial area aggregations aka. geo-hierarchy in the analysis
- Integrate the existing OLAP functions or a part of them with geo-hierarchy
- Provide a seamless integration between temporal data visualization along with its positional or locational information

Handling temporal patterns

- Show chart and map simultaneously
- Overlay charts on map
- Allow user to configure chart parameters
- Provide field names in menu options on the fly once user has uploaded input data
- Pre-generate charts using user-given data and user-configured chart parameters

Handling data-aggregation across geo-hierarchies

- Compute data aggregation on field names based on user-defined geo-hierarchy
- Create a linking between attribute data and geographical data
- Visualize patterns of neighboring geometries simultaneously
- Display different geographical extents (boundary files) on change in geo-hierarchy level
- Display of aggregated or dis-aggregated data based on change in geo-hierarchy level
- Re-generate charts to display this aggregated data

3.4 System Requisites

Hardware requirements The system should have a multi-core processor, operating system installed, browser, sufficient RAM memory, one chart server (phantomjs) and one apache tomcat server. The web sockets should be listening on ports and should be open for communcation. Storage capacity (memory) should be 4 to 10 times of the required size of data storage. The server should have decent bandwidth for quick transfer.

Software requirements Webserver should be able to link to chart server. Additionally, the following softwares need to be installed including PHP, Python (> 2.7), PHP/Java Bridge, Shapely, GDAL/OGR and ogr2ogr.

3.5 Proposed Architecture

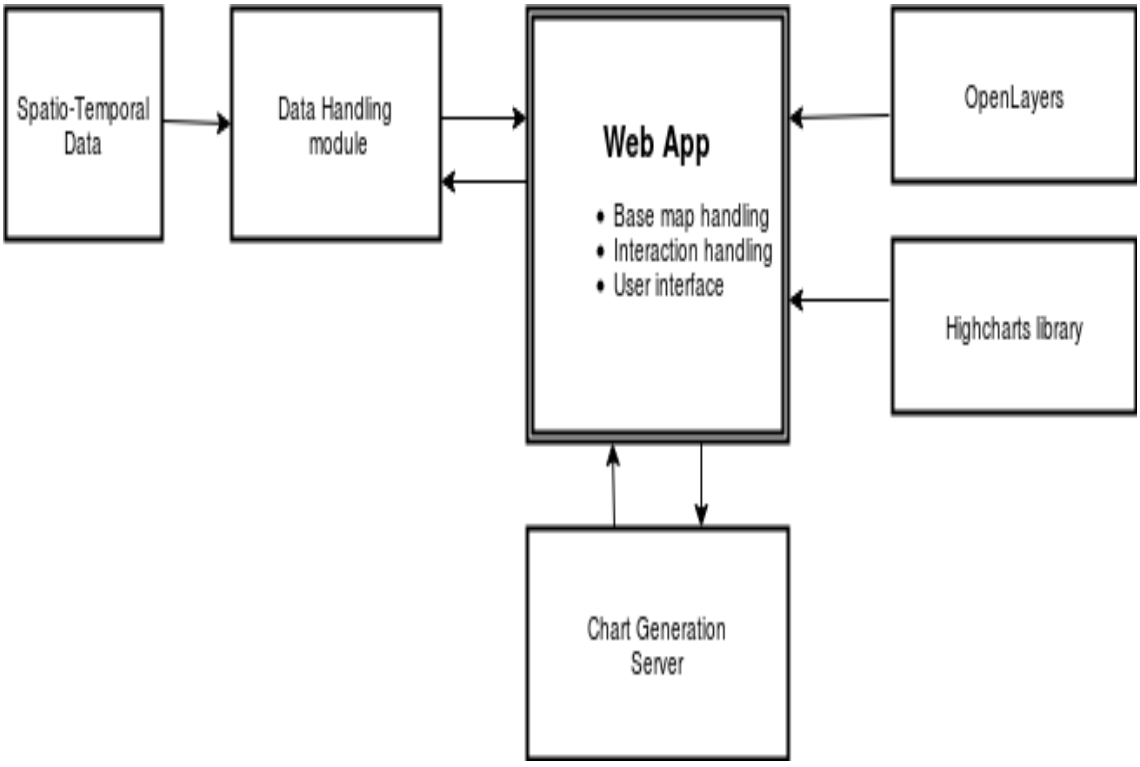

Figure 3.1: Overall Architecture

Figure 3.1 shows the overall architecture of the system. Data handling module is responsible for storing and processing user uploaded data. The web application is hosted on Apache Tomcat 8. OpenLayers library is used to display map on the web page. Charts are configured using the Highcharts JavaScript charting library and before visualizing them on the web application. Phantomjs is the chart generation server. It exports charts on receiving chart generation requests from the web application.

3.6 Working flow of the system

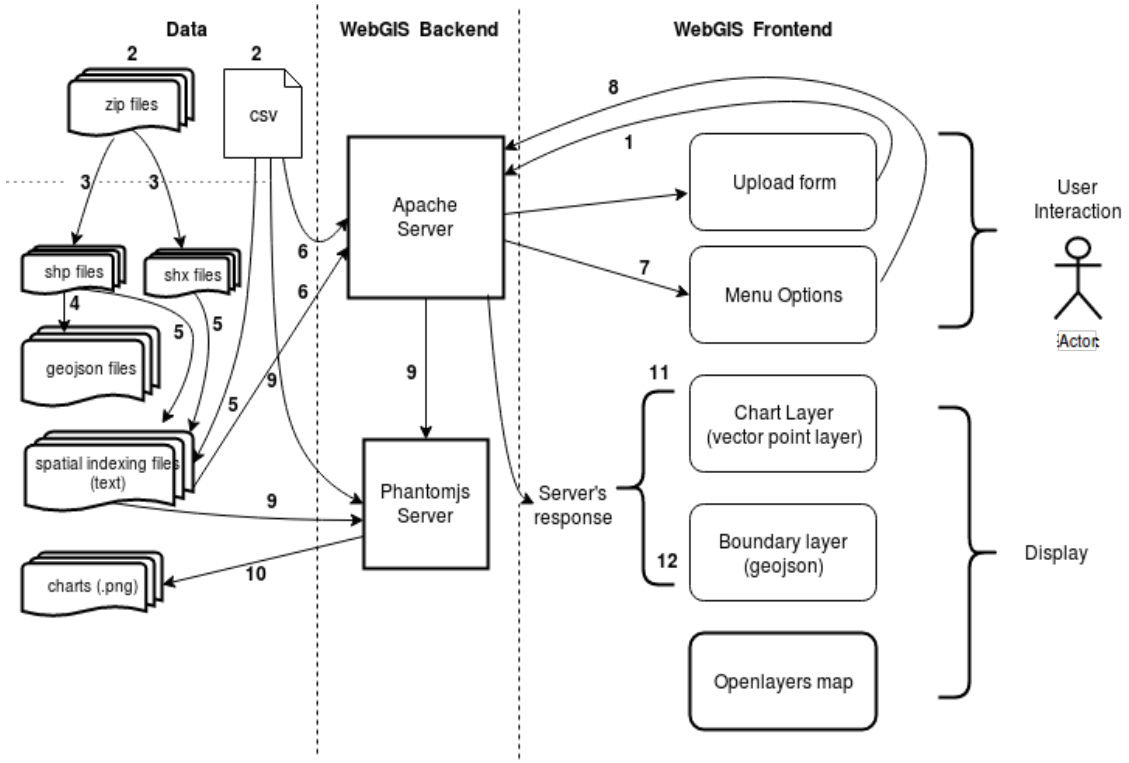

Figure 3.2: Working flow of the system

Figure 3.2 shows the overall flow of the system. The system can be divided into sections - WebGIS Backend, WebGIS Frontend and Data sections. The WebGIS Backend consists of backend processing components including Apache tomcat server and Chart generation server (Phantomjs). The Data section consists of user uploaded input data files, pre-generated charts, spatial model etc. The WebGIS Frontend section has interactive controls for the user and layers to be displayed on map. Chapter 4 describes each of the above components in greater detail.

3.7 Technology chosen

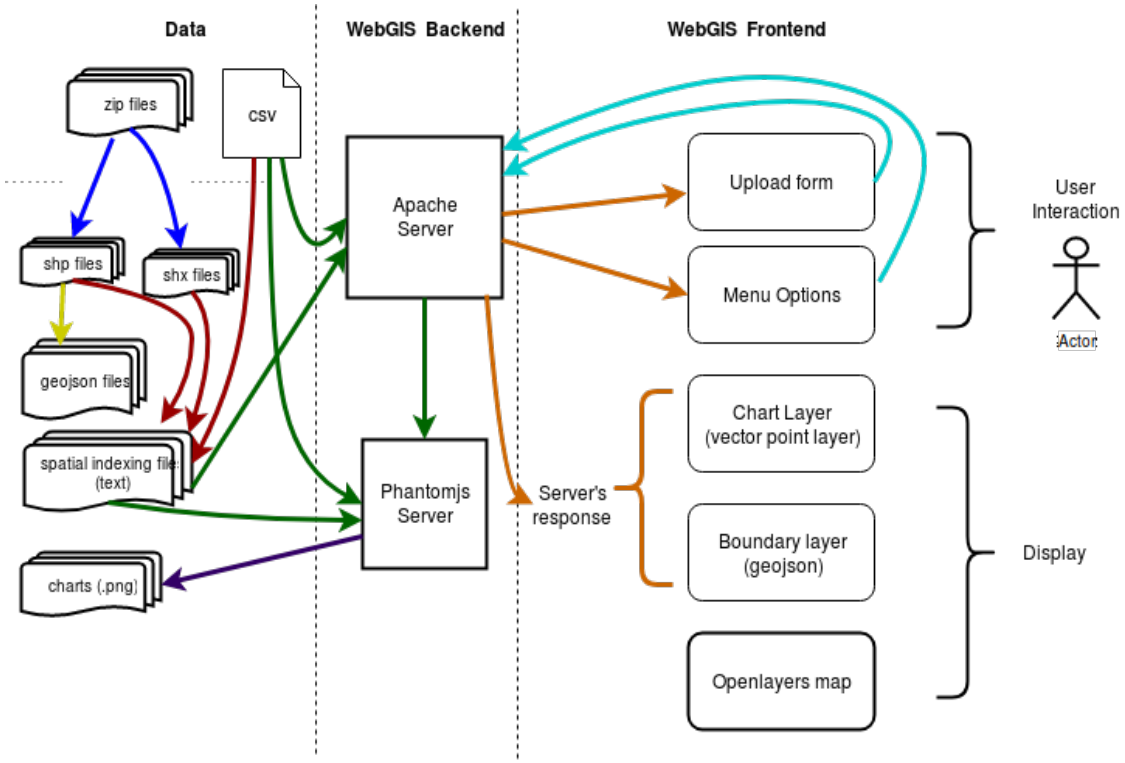

Figure 3.3: Different technologies involved in its implementation represented by different colors

Different colors in figure 3.3 represent the various technologies used in the platform. Table 3.1 gives the list of technologies corresponding to the colors to figure 3.3.

Color	Technologies
Blue	PHP
Red	Python, GDAL, Shapely
Yellow	ogr2ogr
Green	JavaScript, AJAX
Purple	Python, HighCharts
Orange	HTML, JavaScript, CSS
Cyan	JavaScript, jQuery

Table 3.1: Legend

The developed tool is based on a suite of current technologies:

3.7.1 Backend Technologies

3.7.1.1 OpenLayers 2 (web mapping service)

Openlayers is an open source javaScript library which helps to add dynamic content on a map easily. It is completely free. It provides an API for building rich web-based geographic applications similar to Google Maps and Bing Maps [45]. It also supports GeoRSS, KML (Keyhole Markup Language), Geography Markup Language (GML), GeoJSON and map data from any source using OGC-standards as Web Map Service (WMS) or Web Feature Service (WFS).

OpenLayers was created by MetaCarta after the O'Reilly Where 2.0 conference [46] of June 2930, 2005 [47], and released as open source software before the Where 2.0 conference of June 1314, 2006, by MetaCarta Labs.

It includes two renderers - Canvas and WebGL. Both of them support raster data from tile/image servers and vector data [11] .Both Canvas and WebGL can be used only on devices and browsers which support them.

Openlayers 2 supports a range of commercial and free title sources, all projections, most popular vector open source vector data formats, ability to rotate or animate maps, display markers from any source and many others [11].

Simple example of openlayers with OpenStreetMap Layeris given in listing 3.1.

Listing 3.1: Openlayers example

```
<!DOCTYPE HTML>
<title>OpenLayers Example</title>
<div id="simpleMap" style="height:500px"></div>
<script src="OpenLayers.js"></script>
<script>
   map = new OpenLayers.Map("simpleMap");
   map.addLayer(new OpenLayers.Layer.OSM());
   map.zoomToMaxExtent();
</script>
```

OSM is a collaborative project to create a free map of the world. It is built by a community of volunteer mappers who contribute and maintain spatial data.

Compared to Google Maps, you have more flexibility. You can use choose your own map provider and technology. Incase of change in the map provider or technology, you dont have rewrite your entire code. A large number of options are provided including Google, Yahoo, Microsoft, WMS, ArcGIS Server, MapServer. Moreover, you can combine maps from more than one source. It has better support

for vector data types when compared to Google Maps. Debugging is much easier through open source code. You can add new features you need. Also, more styling options are provided.

3.7.1.2 Apache Tomcat 8 (hosting server)

Apache Tomcat is developed by the Apache Software Foundation (ASF). It is an open source Java Servlet Container which implements Java EE specifications and provides a pure Java HTTP web server environment for running Java code [48].

It is released under the Apache License 2.0 license and is an open-source software. Numerous large-scale web applications can be deployed using it.

Apache Tomcat is the world's most widely used web application server, with over one million downloads per month and over 70% penetration in the enterprise datacenter [49]. It can be used from simple one server sites to large enterprise networks [49].

PHP code can be used along with Java application by leaving PHP code in a directory under Apache Document Root and excluding requests to it from being passed to a Java application server like Tomcat. It is helpful in separating Java and PHP applications and running them with only one hosting server and static data like images, CSS, JavaScript. However, integration of both (PHP and Java) is required in certain situations.

PHP is executed by its own separate interpreter which runs PHP scripts to integrate directly into the webserver. To integrate PHP and Java web applications, an easy way is to use PHP/Java Bridge. PHP/Java Bridge is a free and open-source product for such an integration. Moreover, it doesnt require any additional components to invoke Java procedures from PHP.

3.7.2 Scripting Languages

The types of scripting languages used in this system are explained here. More than one scripting languages are being used here so that existing tools and technologies can be seamlessly integrated with it.

3.7.2.1 PHP

Numerous web frameworks exist in a large number of programming languages - PHP, Python, JavaScript, Java, Ruby. Use of a server-side technology was needed to handle it. To choose an appropriate framework for developing our server-side framework, we used the following criteria:

- It should be open-source, under active development, used by a large number of people and have large communities.
- It should allow for integration with multiple technologies.
- It should be fast.

- It should be scalable and reliable.

PHP is a general purpose scripting language especially suited to server-side web development. It was originally created in 1994. PHP has built-in modules for accessing File Transfer Protocol (FTP) and many database servers including MySQL, PostgreSQL, SQLite and Microsoft SQL Server, LDAP servers and others [42].

LAMP architecture includes PHP along with Linux, Apache and MySQL. It is a very popular architecture used in the web industry and is used for deploying web applications.

It is also possible to embed PHP code into HTML, HTML5 markup, and various web frameworks. PHP is usually processed by a PHP interpreter which is implemented as a module in the web server or as a Common Gateway Interface (CGI) executable. The results of the executed PHP code are combined with generated web page by the web server.

The way PHP works is that first code us executed on the server and then the generated HTML is sent to the client. This is different from the way client-side JavaScript works. Here the client only receives the results of running the PHP script and doesnt know the code.

The Hello World! program in listing 3.1 is an example of PHP written in HTML document.

Listing 3.2: PHP written in HTML document

```
<!DOCTYPE html>
<html>
   <head>
      <title>PHP</title>
   </head>
   <body>
      <?php echo '<p>Hello World</p>'; ?>
   </body>
</html>
```

PHP also includes object-oriented functionality. Further, designing, developing and deploying platform using PHP is easy. Hence, based on the above, we selected PHP for platform development.

3.7.2.2 PHP/Java Bridge

The PHP/Java Bridge is an implementation of a streaming, XML-based network protocol which can be used to connect a native script engine with a Java virtual machine. It is much faster than local RPC via SOAP and requires less resources on the web-server side. It is more reliable than direct communication via the Java Native Interface and also doesn't require any additional components to invoke Java proce-

dures from PHP or PHP procedures from Java. J2EE back end clustering and Apache load balancing are supported as well as running PHP scripts within JSP, JSF or other frameworks.

The frontend has PHP whereas backend has Java. The frontend is associated with the backend. To invoke Java procedures and methods, the frontend uses HTTP/XML. A continuation is established between the frontend and the backend. The JavaBridge.war is a distributable ZIP archive which consists of PHP scripts, the JavaBridge.war Java library and PHP Java class. Php-servlet.jar and php-script.jar are included in PHP/Java Bridge web application.

Listing 3.3 shows how PHP access local backend.

Listing 3.3: PHP accessing local backend

```
<?php
require_once("http://localhost:8080/JavaBridge/java/Java.inc");
$System = java("java.lang.System");
echo $System->getProperties();
?>
```

The Java Application Server backends supported are BEA WebLogic, Sun/Oracle Glassfish, Apache Geonimo, IBM WebSphere, Caucho Resin, RedHat JBoss, Sun Java System Application Server, Apache Tomcat and Mortbay/Eclipse Jetty. The supported HTTP servers front-ends are Apache, IIS etc. The PHP/Java Bridge is pure Java JEE application and so user needs Java 1.4 or above running on any operating system or architecture. PHP 4 supported an extension for combining PHP with Java, but to combine PHP with Java in PHP 5 or PHP 6, PHP/JavaBridge is needed.

3.7.2.3 HTML

HTML (Hypertext Markup Language) is that standard markup language for creating web pages and web applications. It describes the structure of a webpage semantically and for the appearance of the document. HTML pages are made up of HTML elements. Using HTML constructs, images and other objects may be embedded into the rendered page. It creates structured documents by denoting structural semantics for text such as headings, paragraphs, lists, links, quotes and other items. HTML elements are delineated tags written using angle brackets. An HTML tag may include other tags as sub-elements. Browsers use HTML tags to interpret the content of the page but they dont display them.

It is used widely to develop web pages. The first standard HTML specification is HTML 2.0. It was published in 1995. HTML 4.0 is a major version of HTML. HTML-5 version is an extension of HTML 4.0.

3.7.2.4 CSS

CSS stands for cascading stylesheets. It is used to style the content and the layout. With the help of CSS, users can change font, color, size, spacing of your content, splitting into multiple columns, adding animations and other features. This improves content accessibility, flexibility, and control over the specification of the presentation characteristics. It enables multiple HTML pages to share formatting by specifying relevant CSS in a separate .css file in order to reduce complexity and repetition in the structural content.

3.7.2.5 AJAX

AJAX stands for Asynchronous JavaScript and XML. It is not a programming language and it uses a combination of a browser built-in XMLHttpRequest object to request data from a web browser and JavaScript and HTML DOM to display or use the data. It allows web pages to be updated asynchronously by exchanging data with a web server behind the scenes. With this, it is possible to update parts of a web page without reloading the whole page.

Web applications send and retrieve data asynchronously with AJAX. With this, the contents on a web page can be dynamically changed without reloading it. AJAX is a group of technologies. HTML and CSS can be used together with style information and markup. The user can interact with new information on web page and modify display using JavaScript.

The following technologies are incorporated in AJAX:

- HTML (or XHTML) and CSS for presentation
- The Document Object Model (DOM) for dynamic display of and interaction with data
- JSON or XML for the interchange of data, and XSLT for its manipulation
- The XMLHttpRequest object for asynchronous communication
- JavaScript to bring these technologies together

An example of a simple Ajax request using the GET method, written in JavaScript is given in listing 3.4.

get-ajax-data.js:

Listing 3.4: Simple AJAX Request

```
var xhr = new XMLHttpRequest();
xhr.open('GET', 'send-ajax-data.php');
xhr.onreadystatechange = function () {
   var DONE = 4;
```

```
   var OK = 200;
   if (xhr.readyState === DONE) {
      if (xhr.status === OK) {
         console.log(xhr.responseText);
      } else {
         console.log('Error: ' + xhr.status);
      }
   }
};
xhr.send(null);
```

Simple PHP script is given in listing 3.5.
send-ajax-data.php:

Listing 3.5: PHP script

```
<?php
header('Content-Type: text/plain');
echo "This is the output.";
?>
```

It is very intuitive and based on natural user interaction. Clicking is not required, mouse movement is a sufficient event trigger. It is more data-driven as opposed to page-driven.

3.7.2.6 jQuery

jQuery is a lightweight JavaScript library which makes it easier to use JavaScript on your website. It wraps a lot of common tasks involving many lines of code into methods which can be called using single line of code. Listing 3.6 gives jQuery example.

Listing 3.6: jQuery example

```
$(document).ready(function(){
  // jQuery methods
});
```

All jQuery methods in our examples, are inside a document ready event. This is to prevent any jQuery code from running before the document is finished loading (is ready). If methods are run before the

document is fully loaded, actions such as trying to hide an element that is not created yet and trying to get the size of an image that is not loaded yet can fail.

jQuery's syntax is designed to make it easier to navigate a document, select DOM elements, create animations, handle events, and develop AJAX applications.

- Elimination of cross-browser incompatibilities: The JavaScript engines of different browsers differ slightly so JavaScript code that works for one browser may not work for another.

The list of important core features supported by jQuery:

- DOM manipulation The jQuery made it easy to select DOM elements, negotiate them and modifying their content by using cross-browser open source selector engine called Sizzle.
- Event handling The jQuery offers an elegant way to capture a wide variety of events, such as a user clicking on a link, without the need to clutter the HTML code itself with event handlers.
- AJAX Support The jQuery helps you a lot to develop a responsive and feature rich site using AJAX technology.
- Animations The jQuery comes with plenty of built-in animation effects which you can use in your websites.
- Lightweight The jQuery is very lightweight library - about 19KB in size (Minified and gzipped).
- Cross Browser Support The jQuery has cross-browser support, and works well in IE 6.0+, FF 2.0+, Safari 3.0+, Chrome and Opera 9.0+
- Latest Technology The jQuery supports CSS3 selectors and basic XPath syntax.

Listing 3.7 gives example of jQuery embedded with HTML.

Listing 3.7: jQuery embedded with HTML

```
<html>
  <head>
    <title>The jQuery Example</title>
    <script type = "text/javascript" src = "/jquery/jquery-2.1.3.min
       .js">
    </script>
    <script type = "text/javascript">
      $(document).ready(function() {
        document.write("Hello, World!");
```

```
      });


    </script>
  </head>
  <body>
    <h1>Hello</h1>
  </body>
</html>
```

3.7.2.7 JavaScript

Javascript or JS is a high-level, dynamic, weakly-typed, and interpreted programming language. It is commonly used for making webpages interactive and provide online programs. It is one of the core technologies for World Wide Web content production. Using it as a built-in Javascript engine in modern web browsers eliminates the need for plug-ins. Previously, it was used only on the client-side but today it is being used in server-side implementations in web servers and databases as well as on client-side.

Netscape communications wanted a scripting language which would complement Java and also have a similar syntax without adopting other language such as Perl, Python, TCL or Scheme. It was first developed under the name Mocha and was officially called LIveScript. Later in September 1995, it was renamed JavaScript. Till December 1995, it was released only for browsers. After that, it was introduced for server-side scripting.

Today, it is one of the most popular programming languages on the web. With the arrival of AJAX, it has received even more attention. This led to an improvement in JavaScript programming practices and increased its usage on server-side platforms. Basically, when JavaScript code is run locally in a users browser, it makes the application more responsive by making the browser respond to user actions quickly.

Example of a web page containing JavaScript is given in listing 3.8.

Listing 3.8: JavaScript example

```
<html>
  <head>
    <title>Sample</title>
  </head>
  <body>
```

```
   <button id="hi">Hi</button>
   <script>
      document.getElementById('hi').onclick = function() {
         alert('Hi!');
         var simpleTextNode = document.createTextNode('Some Text
            ...');
         document.body.appendChild(simpleTextNode);
      };
   </script>
  </body>
</html>
```

JavaScript is supported today in all modern browsers with built-in interpreter.

3.7.2.8 Python

Python has a very simple structure and clearly defined syntax. It is a general-purpose interpreted, interactive, object-oriented and high-level programming language. It was created by Guido van Rossum during 1985- 1990 at the National Research Institute for Mathematics and Computer Science in the Netherlands. Python libraries are compatible with all commonly used operating systems UNIX, Windows and Macintosh. It also provides interfaces to all major databases. It is possible to write python code using both GUI and command line tools. It is also scalable. It provides structure and support for users to write large programs. Apart from this, python can be run on a wide variety of hardware platforms and also has support for interactive debugging and testing. It has efficient high-level data structures.

Python is free and open-source. Anyone can freely use and distribute Python, even for commercial use. Not only can anyone use and distribute softwares written in it, he can even make changes to the Python's source code. Python has a constantly growing large community.

Python is extensible and embeddable. User can combine python code with pieces of C/C++ or other languages so that the application can give high performance and scripting capabilities.

Apart from the above-mentioned features, python also supports functional programming methods. It supports filter(), map() and reduce() functions.

It supports dynamic type checking and high-level dynamic data types. There is automatic garbage collection feature too. It can easily integrated with C, C++, COM, ActiveX, COBRA, and Java.

The design began in the late 1980s and was first released in February 1991. In late 1980s, Guido Van Rossum was working on the Amoeba distributed operating system group and wanted a simple easy-to-understand language which could access Amoeba system calls. This led to its design.

The document The Zen of Python (PEP 20) which summarizes the languages core philosophy includes aphorisms such as explicit is better than implicit, simple is better than complexa among others.

Rather than having all of its functionality built into its core, Python was designed to be highly extensible. Therefore, it has compact modularity.

3.7.3 Key Components

3.7.3.1 Highcharts javascript library

Highcharts ia an open-source tool for generating javascript-based charts. It is fast, flexible and can be extensively customized [50]. A user can making his own edits after downloading the source code. It also provides flexible styling options. It has a vibrant community on Github, Stackoverflow and other forums. So, user can get additional assistance and implementation advice. It supports multiple data input formats including CSV, JSON and multiple backend database or server stacks. Apart from this, it provides wrappers in all popular languages such as .Net, PHP, Python, R, Java and iOS.

With a charting library that supports all types of charts, charts can be made easily. Of the popular charting libraries present, Highcharts library was chosen mainly because it can be integrated along with phantomjs chart export server being used. Moreover, the library supports a huge number of chart types, has maximum styling options for a chart, data to be displayed can be chosen, among others.

For different types of charts, different constructors need to be called. For example- for stock chart, Highcharts. StockChart method should be called. The data is added in JavaScript array using a separate JavaScript file or an AJAX call to the server. The code in listing 3.9 is an example of stockChart.

Listing 3.9: Highcharts example

```
var chart;
$(function() {
     chart = Highcharts.stockChart('container', {
     rangeSelector: {
         selected: 1
       },
       series: [{
         name: 'INR to USD',
         data: inrtousd // predefined JavaScript array
       }]
```

```
    });
});
```

The code in listing 3.10 is an options object which can be added to the chart by passing it to the chart constructor.

Listing 3.10: Highcharts Chart constructor

```
$(document).ready(function() {
   var chart = new Highcharts.Chart(options);
});
```

In addition to this, new members can be easily added by pushing it into the series. Listing 3.11 shows how new members are added to a Highcharts chart.

Listing 3.11: Add new members to Highcharts chart

```
options.series.push({
   name: 'John',
   data: [3, 4, 2]
});
```

3.7.3.2 Shapely (for server-side scripting)

It is not concerned with data formats or coordinate systems, but can be readily integrated with packages Python and GEOS packages. For example, Shapely 1.6 require Python version greater than or equal to 2.6 and GEOS version to be greater than or equal to 3.3.

It manipulates and analyzes data easily since it is based on GEOS which is the standard library for those tasks and is very fast. With Shapely, user can easily make buffers, unions, intersections, centroids, convex hulls etc. quite efficiently.

Shapely enables python programmers to perform PostGIS type geometry operations outside RDMS. If there is no mandate to manage data over time in the database, shapely can be used to do work. With shapely, hundred different GIS format readers and writers and the large number of state plane projections are not required.

Shapelys purpose is a generic geometry library. It is a high-level, pythonic interface to the GEOS library for geometry operations. It avoids reading or writing files and relies completely on data import and export. Its main focus is on geometry manipulation. Listing 3.12 shows example of shapely.

Listing 3.12: Shapely example

```
from shapely import wkt, geometry
wktPoly=POLYGON((0 0, 4 0, 4 4, 0 4, 0 0))
poly = wkt.loads(wktPoly)
poly.area
16.0
buf= poly.buffer(5, 0)
buf.area
174.4137122636848
```

3.7.4 Additional Libraries

3.7.4.1 GDAL/OGR

The Geospatial Data Abstraction Library (GDAL) is a computer software library for reading and writing raster and vector geospatial data formats, and is released under the permissive X/MIT style free software license by the Open Source Geospatial Foundation. As a library, it presents a single abstract data model to the calling application for all supported formats. It may also be built with a variety of useful command line interface utilities for data translation and processing. Projections and transformations are supported by the PROJ.4 library. The related OGR library (OGR Simple Features Library), which is part of the GDAL source tree, provides a similar ability for simple features vector graphics data. It also comes with a variety of useful command line utilities for data translation and processing.

3.7.4.2 ogr2ogr

Ogr2ogr is a utility that comes witha library which can read OSM data and write formats supported by OGR. FW Tools toolkit has a subkit FW Tools Toolkit which has several command line options. OgrInfo and Ogr2Ogr are found very useful. The OgrInfo gives a summary data and detailed information about the layers and geometries. Ogr2Ogr converts one Ogr defined source to another data source. Ogr supports multiple data formats including ESRI Shapefile, PostGIS etc.

3.8 Development Environment

The development platform for the proposed framework is a PC with Intel Core 2.5GHz x4 CPU, 7.7GB RAM, and Ubuntu 16.04. The web page is deployed in Tomcat 8.0 server. The web browser is

Mozilla Firefox. The Web Socket is deployed in port 8080. All tests and experiments are carried out in the local host.

Chapter 4

LSI-STAT and its modules

4.1 Platform Components

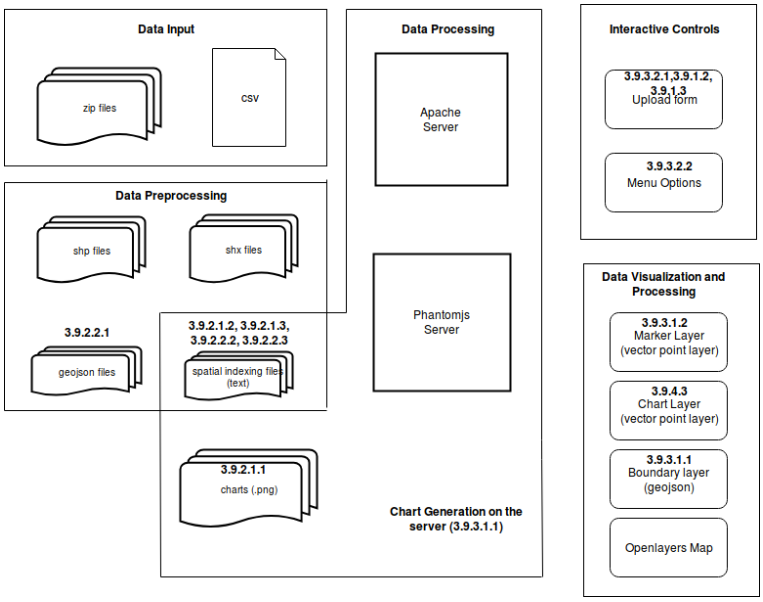

Figure 4.1: Block Diagram with Platform Components

Figure 4.1 is a detailed architecture diagram of the tool. Various components have been divided into blocks - Data Input, Data Preprocessing, Data Processing, Data Processing and Visualization and Interactive Controls. For detailed explanation of each of the components, please refer to the section numbers added in the figure. This figure gets translated into modules and these modules are described below in this chapter.

4.2 Functional Modules

4.2.1 Data Input

This subsection explains working of the data input module of the tool which is responsible for handling the user-uploaded data.

4.2.1.1 Input Data Format

The tool accepts user input data in csv format and boundary file in zip format. Boundary files (in .shp and .shx formats) have a large file size. Therefore, the number of boundary files that the user can upload is restricted to 5. CSV is one the most popularly used text file format (needs support). Therefore, the data file format for the tool is set as csv. The tool assumes that the csv file uploaded by the user is cleaned and verified. No data verification is done by the system. Also, the system doesnt take care of missing data in csv file. Geolocation latitude and longitude coordinates are assumed to be already present in the file. Shapefile (.shp) and ShapeIndex file (.shx) both are required for processing by the tool. So, the input format for boundary files is set as zip format. Sample CSV Input is given in listing 3.1.

Listing 4.1: Sample CSV Input

```
'Mandal',Longitude,Latitude,'PatientCounts','Population','Timetaken
   ','Area','Density','Hospital','Distance','PopIn1hr','Pop1to2hr','
   Pop2hrplus','AreaIn1hr','Area1to2hr','Area2hrplus'
'KHAMMAM (Urban)
   ',80.142925,17.248909,38,313504,6,231.12,559.4236760125,'Mamatha
   ',6,129294,0,0,231.12,0,0
'KONIJERLA',80.292996,17.222976,8,61321,21,233.58,262.5267574279,'
   Mamatha',21,61321,0,0,233.58,0,0
'ENKURU',80.440577,17.330158,0,35342,53,183.83,192.2537126693,'
   Mamatha',53,35342,0,0,183.83,0,0
```

4.2.1.2 User-defined geo-hierarchy

For a single dimension, multiple hierarchies may be defined. Example, for time dimension, day, month, and year is an order. Second order could be day, month, quarter, and year. Third order could be day, fiscal week, fiscal month, fiscal quarter, fiscal year. So, the system gives the user the flexibility to decide and set the hierarchy order based on his requirement and not enforce any particular order

for geographical hierarchy. User has to set this order too, while uploading the boundary shapefiles. The ascending order of the user uploaded the boundary files is set as the geo-hierarchy order from the highest to lowest hierarchy i.e. the first uploaded boundary file would be the highest in the geo-hierarchy. Example of a geo-hierarchy order is Country, followed by State and then District.

4.2.1.3 Side panel design: upload options

Upload options were added to the side panel to allow user to upload input files. jQuery Form Plugin was used for allowing input files in zip format. Using these options, user has to choose the data file (.csv) and then the boundary file (.zip) to be uploaded in the tool. The chosen files are then verified by the tool and moved to the upload folder present on the server machine. Flowchart in figure A.1 represents the steps for uploading user input data.

4.2.2 Preprocessing

This subsection explains the preprocessing module of the tool. Preprocessing certain information required for data visualization greatly increases interactivity.

4.2.2.1 Data based preprocessing

4.2.2.1.1 Preprocessing vs on the fly chart generation Using a pre-processing based approach has several advantages over on the fly based. It increases interactivity by pre-generating charts and storing them in chart cache. So, when the tool requires new set of charts to be added on the map, it simply replaces the old set of charts with new set of charts in a vector layer.

4.2.2.1.2 Point based hierarchy approach For generating charts corresponding to each of the polygons in a shapefile, the data points contained inside each of the polygons needs to be known. This involves checking each of the data points in the user-given data file for each polygon of a shapefile. This involves a lot of processing to be done. Performing this computation on the fly would lead to larger delays and decrease interactivity. Hence, it was decided to pre-compute this information. In order to store this information, two approaches were mainly considered. One of the approaches involved a point-based hierarchy and another used polygonal hierarchy.

In the first approach, for each shapefile, a mapping is created between each polygon and the points that are enclosed inside it and stored in a file. So, the number of such files created is equal to the number of shapefiles uploaded by the user. Since the maximum number of shapefiles that can be uploaded by the user is set to 5, so not more than 5 files need to be created for storing mapping. User-defined geographical hierarchy is used along with this mapping to generate charts for data visualization.

In the second approach, a tree data structure would need to be used to store information about geographical hierarchy between polygons. The root node of this tree would be the shapefile at the highest

geographical level. The number of levels would be equal to the number of geographical hierarchical levels. Since the maximum number of shapefiles that can be uploaded by the user is set to 5, so the tree would have not more than 5 levels. The children of a node would be the polygons of lower geographical level which are contained inside the polygon at that level corresponding to the node.

The major difference between the approaches is that in the second approach information about hierarchy among the polygons of level and another level is stored but not so in the first approach. Both the approaches have their own advantages and disadvantages. To time taken to search the data records to be aggregated for any polygon at a specific geographical hierarchy is easier in point-based model compared to polygon-based. Also, any change in geographical hierarchy order would involve modifying the tree structure in the second approach but there would be no change in any of the mappings in first approach. Considering the above merits of the first approach, point-based hierarchy model was used in this tool.

4.2.2.1.3 Spatial Aggregation procedure To find whether a data point lies inside the polygon, Shapelys *within* geometry predicate was used. For points lying on the boundary of polygons, they were considered to lie in any one of the polygons sharing the boundary. It was ensured that each point was mapped to a single polygon. Another approach for finding whether a point lies inside a polygon is using the ray casting algorithm. However, using Shapelys inbuilt *within* predicate is faster than that approach.

This was to be done for all the data points given by the user. For this, first all the points were sorted on the basis of their values on the x-axis and then on y-axis. Then, for each point, it was checked whether it is contained within in any polygon. A mapping from each polygon to the set of points lying inside it was created. Further, a list of polygon ids containing at least one data point was created. Based on the data, it may happen that majority of the data points are present inside a few polygons. As a result of this, many polygons would have no data values and thus no charts would be required to be generated. Thus, a list of polygon ids was made with charts corresponding to the values of the points mapped to only help generate chart for these polygons. So, there is no need to check all polygons to find whether they contain any points.

4.2.2.2 Map based preprocessing

4.2.2.2.1 Converting shapefile to GeoJSON file For displaying a shapefile on map, it has to be converted to GeoJSON, KML or GML file type and added on map. Out of these, GeoJSON is smaller in size than KML or GML. Therefore, parsing GeoJSON and visualizing on map would take less time. Hence, the shapefile was converted to GeoJSON. This was done using ogr2ogr driver as it is one of the simplest ways to convert the file. GeoJSON files corresponding to choose each shapefile need to be created and stored. The required GeoJSON file is added later on to the openlayers map. Flowchart in figure A.2 represents the steps for preprocessing the user uploaded files.

4.2.2.2.2 Positioning charts inside polygon Each chart was to be placed at the center of the polygon it lies in. For this, the centroid of each polygon was computed. For convex polygons, a representative point lying in the center of its interior region was used in place of the actual centroid.

4.2.2.2.3 Chart size computation Some polygons are larger in size than others. However, for all charts lying inside any polygon, a single chart size was chosen. The rationale behind this was that comparing data values or visualizing trends in charts of same size is easier and if required the user can use the *Zoom all charts* functionality to zoom and view charts as required.

4.2.3 Processing

4.2.3.1 Map based processing

4.2.3.1.1 Layer CRS A large amount of software used the identifier EPSG:900913. This is an unofficial code, but is still the commonly used code in OpenLayers 2. The string EPSG:900913, it will be describing coordinates in meters in x/y. Moreover, if you want to use maps from any other services like WMS, then it can integrated with the existing map without much difficulty. If user wants some kind of metric system for queries to be built in the future, it can be done easily. For this tool, EPSG:900913 was decided as the default projection for displaying all boundary files. The user has to give shapefile with this projection only. If he gives shapefile in any other format, still the display projection would be set as EPSG:900913 for this shapefile. As a result of this, the shapefile would not be displayed correctly on the map. Moreover, the charts displayed would be inaccurate since incorrect data points would have been aggregated. No conversion is being done by this tool. It can be added in future.

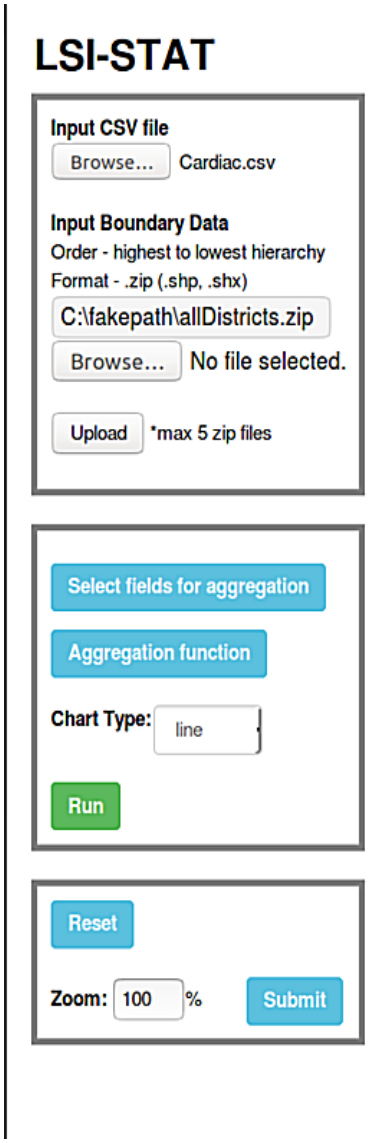

Figure 4.2: User Interface- Side Panel

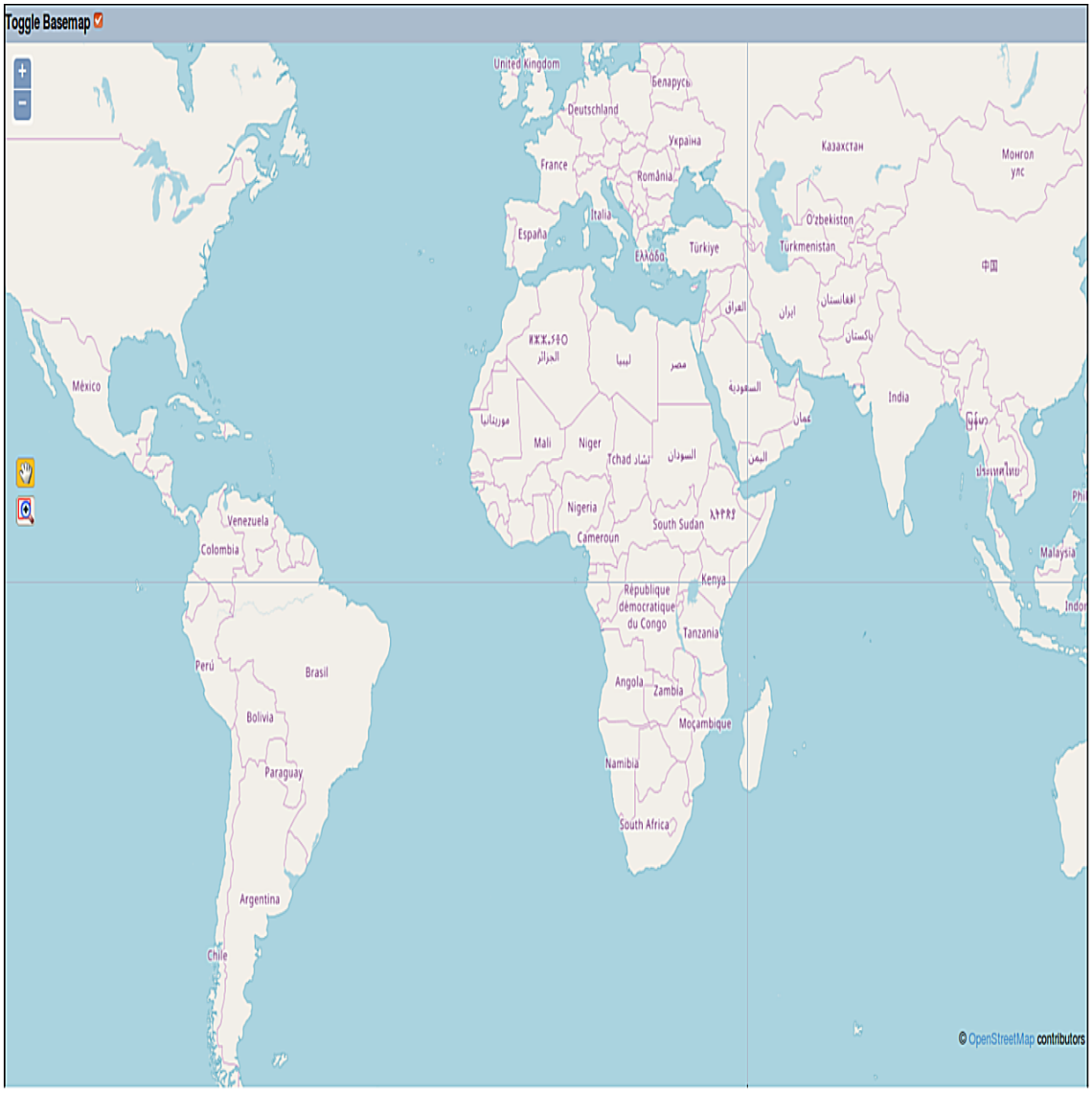

Figure 4.3: User Interface- Map Panel

This tool has two main components. A side panel with different menus is present on the left and a map panel on its right. Figure 4.2 shows side panel. The side panel provides various interactive controls for uploading files and configuring different chart attributes such as the type of chart, attributes to be displayed and aggregation function to be selected. It also has an option for changing the map layer for visualization. Figure 4.3 shows map panel. The map panel displays the generated charts and the boundary layer on an OpenLayers map. It has controls which allow the user to zoom in, zoom out or shift the map towards left or right.

4.2.3.1.2 Initial setup Once the initial setup is complete, markers for all data points are displayed on the OpenLayers map in the map panel of the tool. Data visualization helps the user to get initial overview

of the geographic extents of the entire dataset and geo-coordinates of all data records. Flowchart in figure A.3 represents the steps for initial setup of map panel for geovisualization.

4.2.3.2 Data based processing

4.2.3.2.1 Interactive Controls

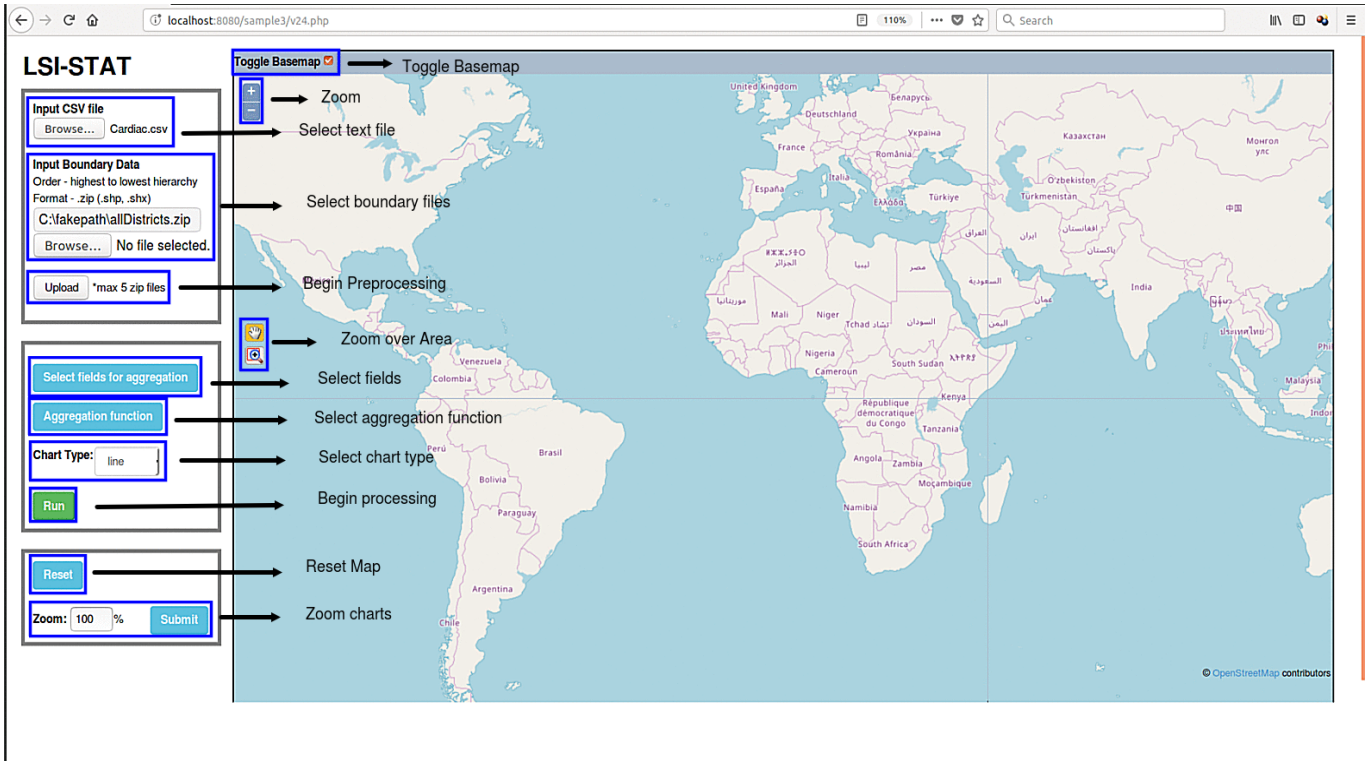

Figure 4.4: Interactive Controls

Figure 4.4 shows the various interactive features present in the platform, They have been explained below.

Select Attributes

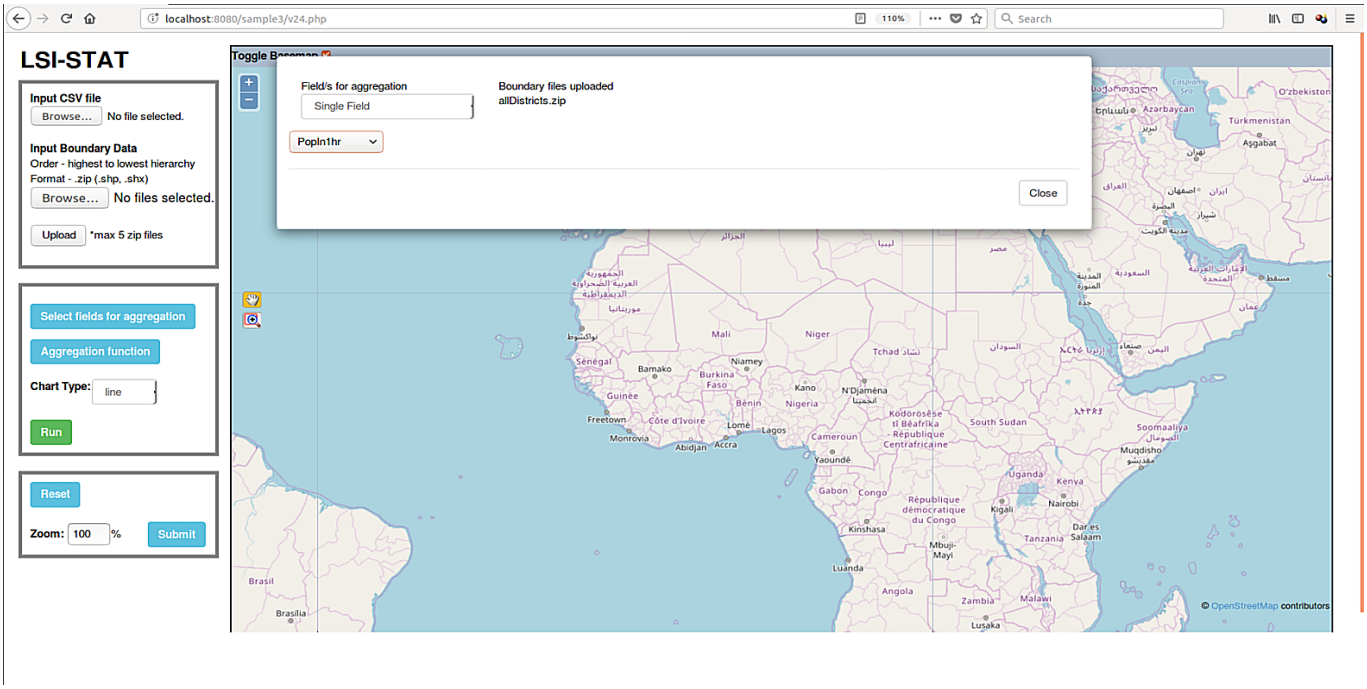

Figure 4.5: Select Attributes - Single Field

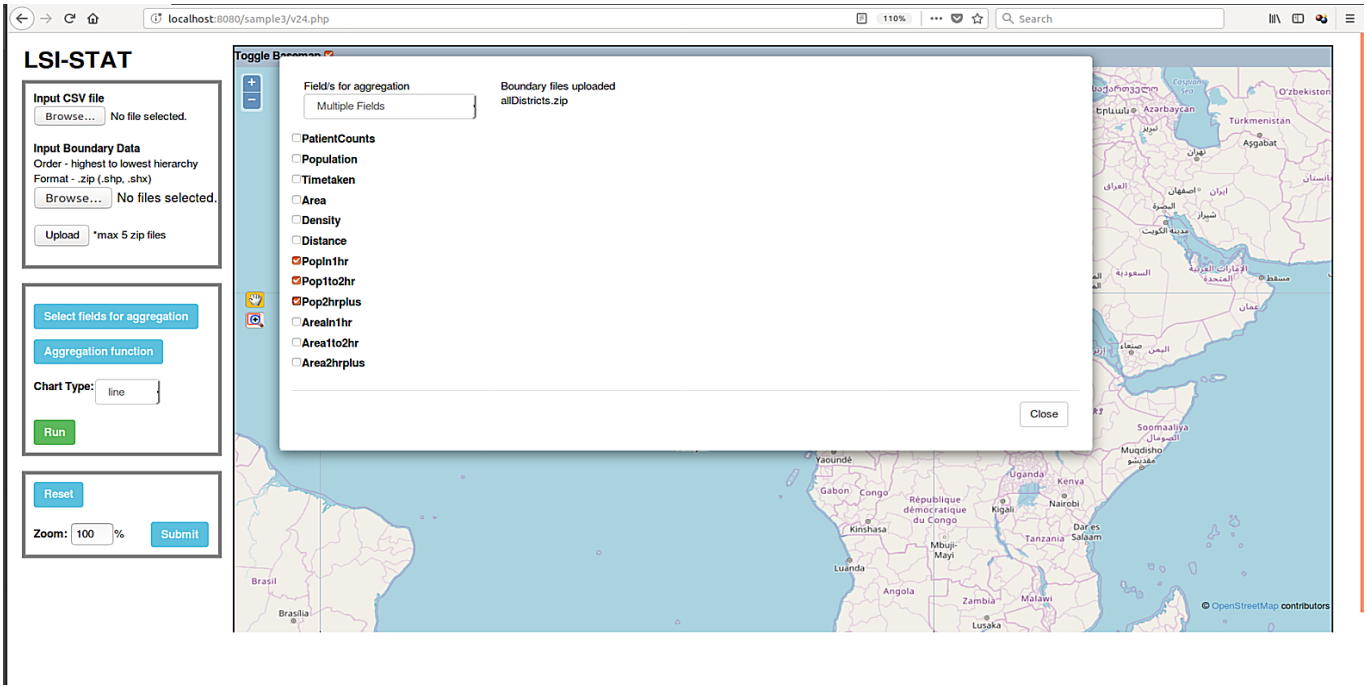

Figure 4.6: Select Attributes - Multiple Fields

In order to discover patterns and relations in the data, a user should be able to explore one or more data attributes. A menu on the side panel of the tool allows the user to select the attributes to be displayed on the chart. Data values from the selected attributes are later sent to phantomjs server for chart generation. Single or multiple fields can be selected for this. Figures 4.5 and 4.6 show selection of single and multiple attributes respectively to be displayed by user. The steps involved have been shown in flowcharts. Flowchart A.4 show the steps involved in invoking relevant functions based on user chosen option. Flowchart A.5 represents steps involved in extracting data from user input file for providing fields in menu. Flowchart A.6 represents steps for providing fields in dropdown menu. Flowchart A.7 shows steps for providing fields as checkboxes.

Aggregation type

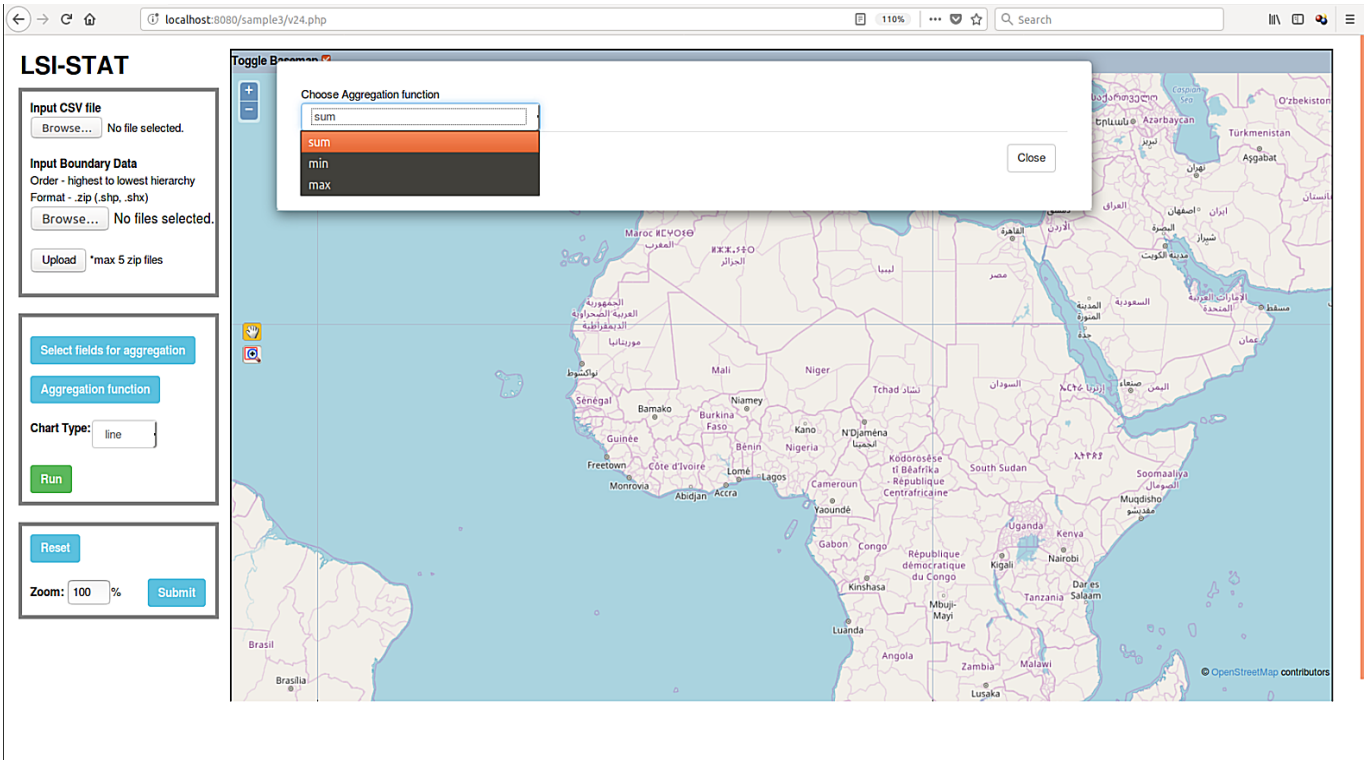

Figure 4.7: Aggregation Type Selection

Using this menu option, the user can select the aggregation function to be applied on the data values of the user-selected attributes. Different aggregation functions present are sum, max and min. Figure 4.7 shows selection of aggregation type by user. Flowchart A.8 gives the steps for fetching user selected option and displaying field names.

Chart Type

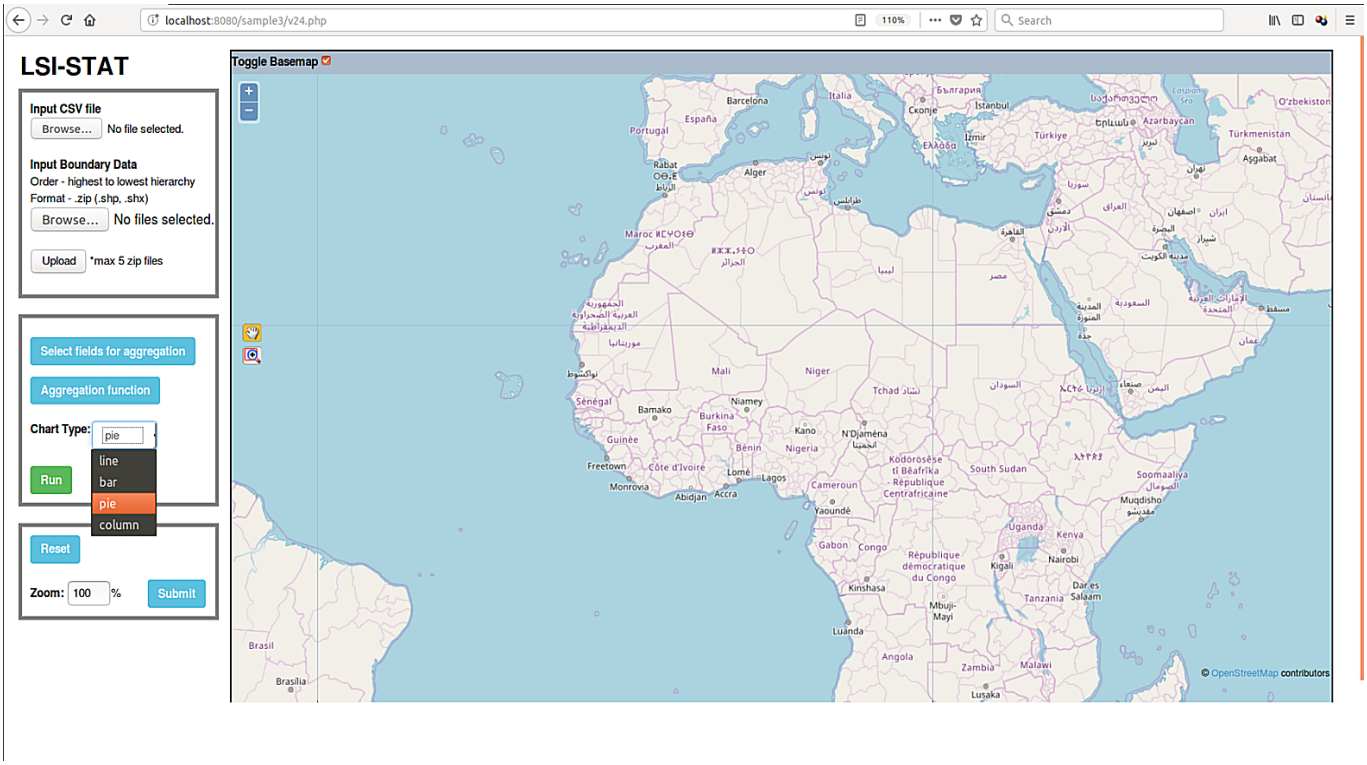

Figure 4.8: Chart Type Selection

Using this feature, the user can select the type of chart (bar, pie, line) in which he wants to represent data. Figure 4.8 shows this. Different chart types are suitable for different data. It depends on data characteristics as well as the type of data (geographical, sales, health data) being used. Hence, the tool lets the user to decide and select the best chart type for this data.

Reset

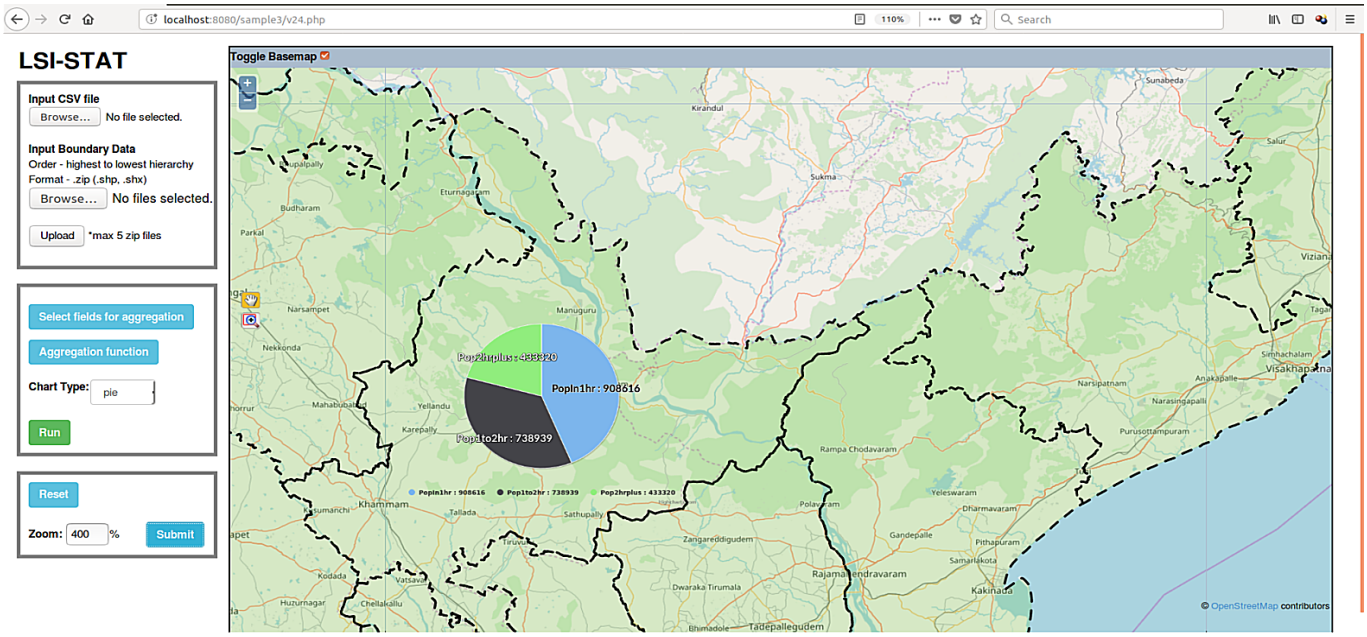

Figure 4.9: Before Reset

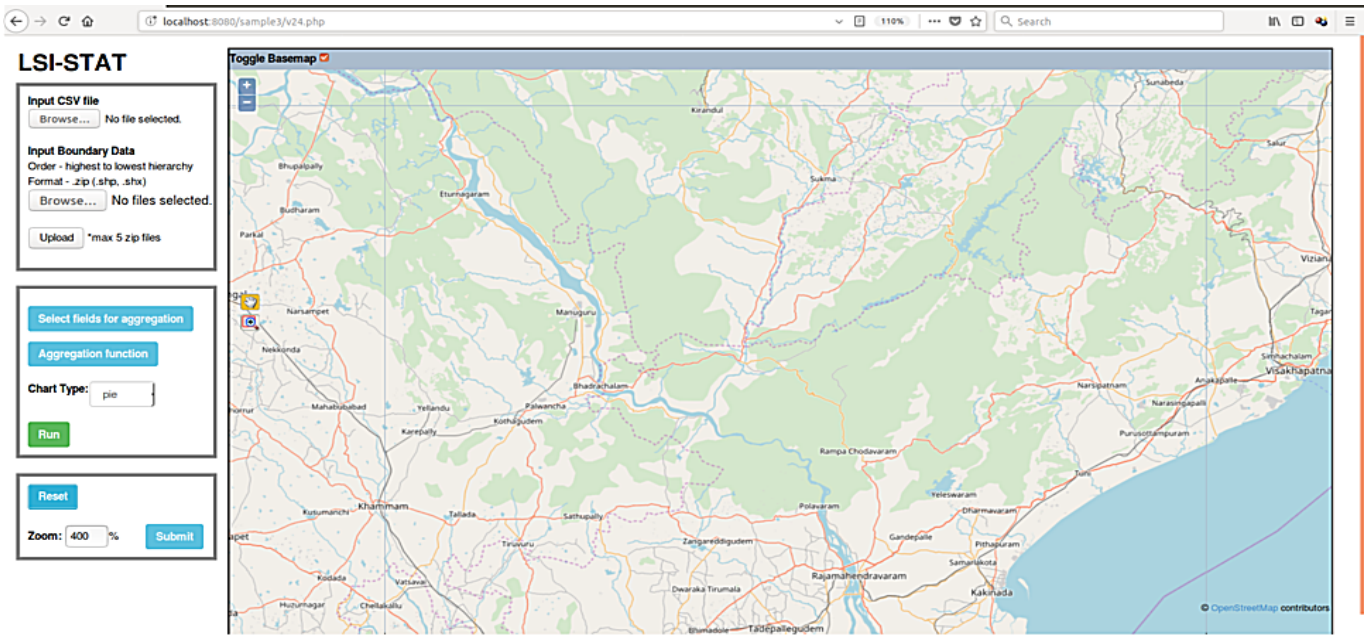

Figure 4.10: After Reset

Using this feature, the users can reset the map. Reset here refers to removing all layers present on the map. Figure 4.9 shows initial map with chart and boundary layers and 4.10 shows the map without any layers.

Zoom all charts

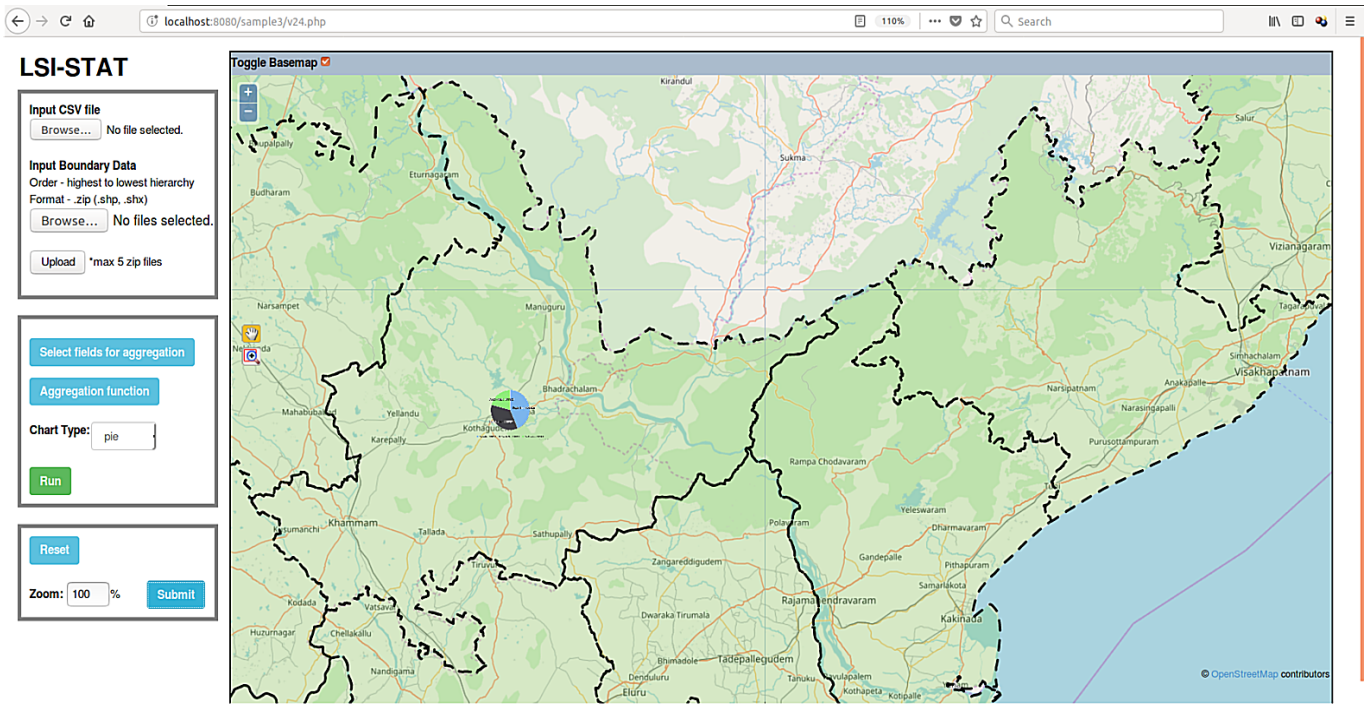

Figure 4.11: Initial Map with charts

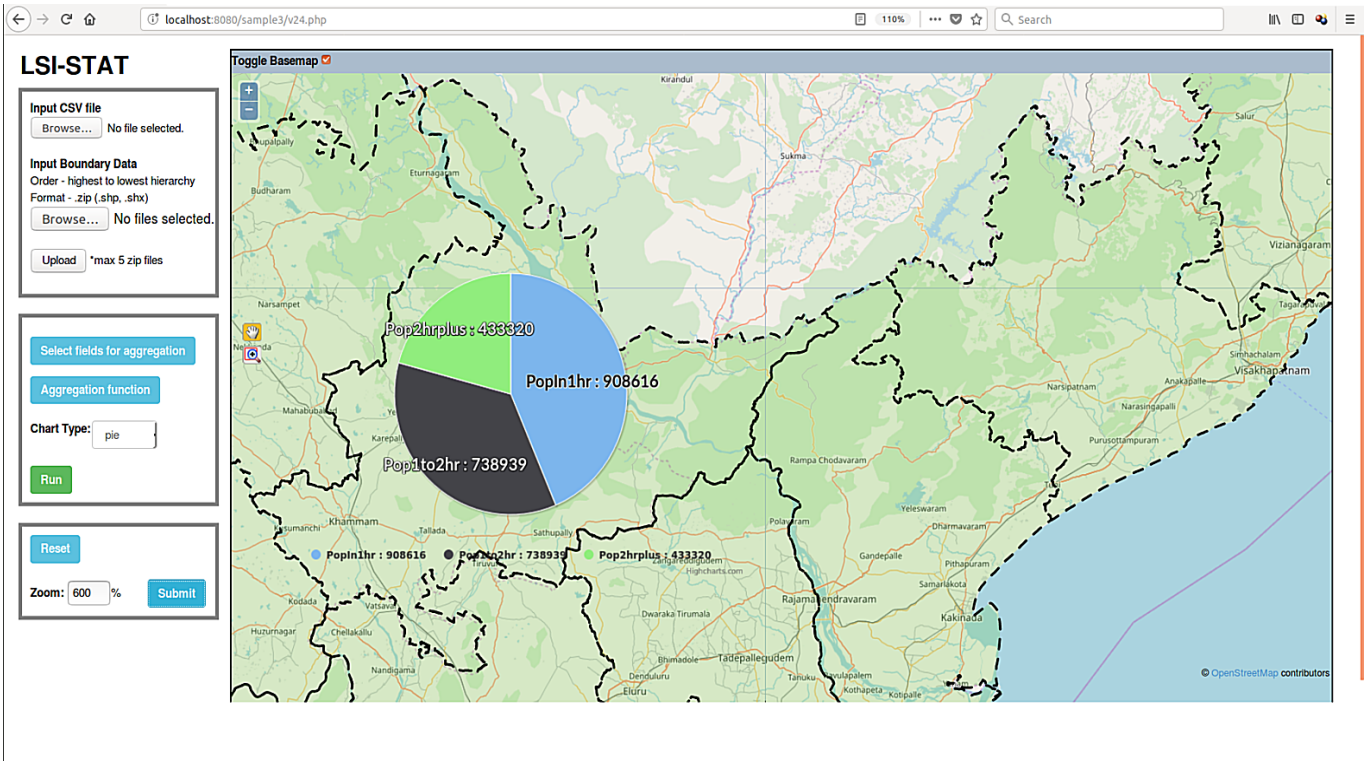

Figure 4.12: Zoom all charts by 600%

Using this feature, the user can zoom all charts by a fixed percentage. User has to enter the percentage in the text box and click submit. All charts will be zoomed by the percentage entered by the user. Initial chart size for all charts is set to 50px. Figure 4.11 shows initial map with charts and Figure 4.12 shows map when all charts are zoomed by 600%. Figure A.9 shows steps for zooming all charts by a user-defined percentage.

Toggle Basemap

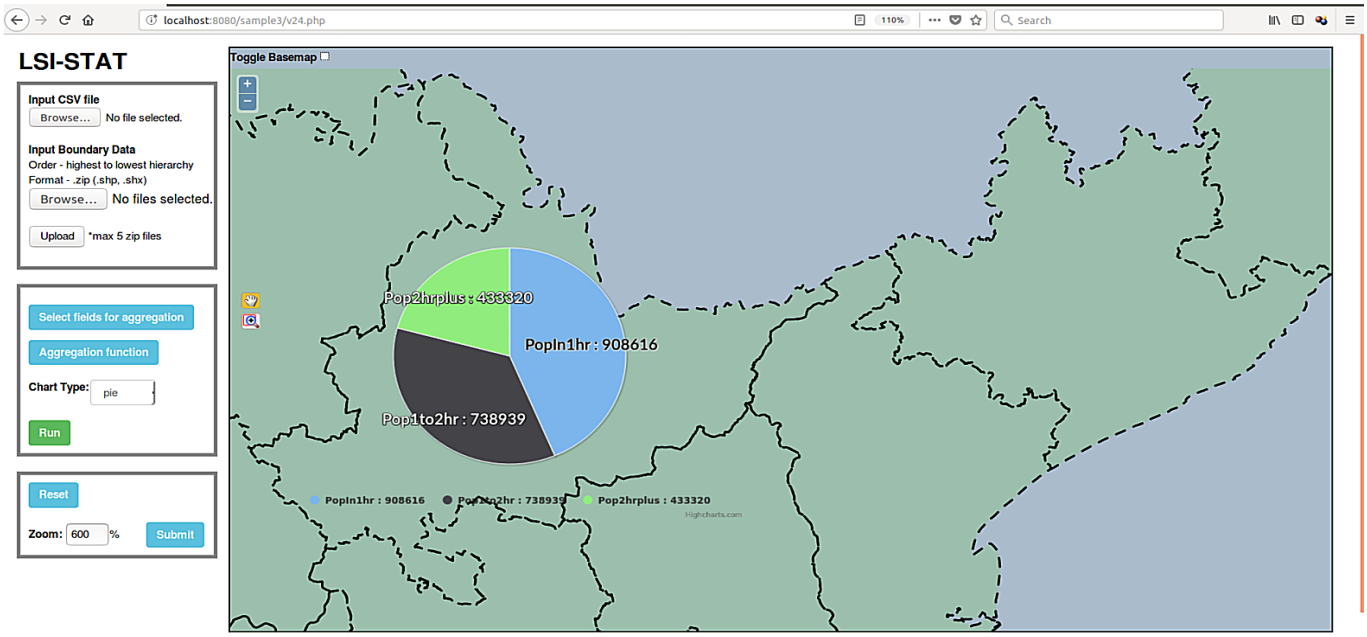

Figure 4.13: Map without openlayers basemap

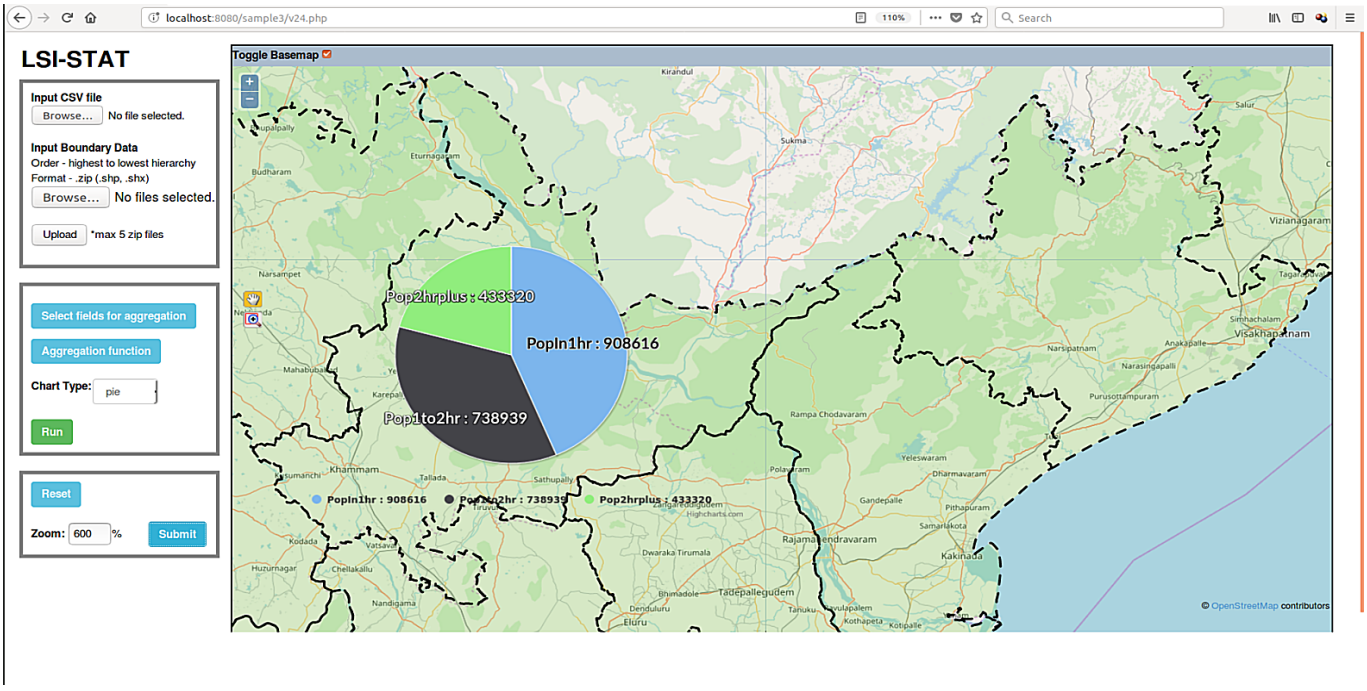

Figure 4.14: Map with openlayers basemap

Figures 4.13 shows map without openlayers as basemap and 4.14 shows map with openlayers as basemap. It is present as a checkbox in map panel of the tool. Flowchart in figure A.10 shows the steps for toggling the basemap.

Zoom by Area and Shift left/right

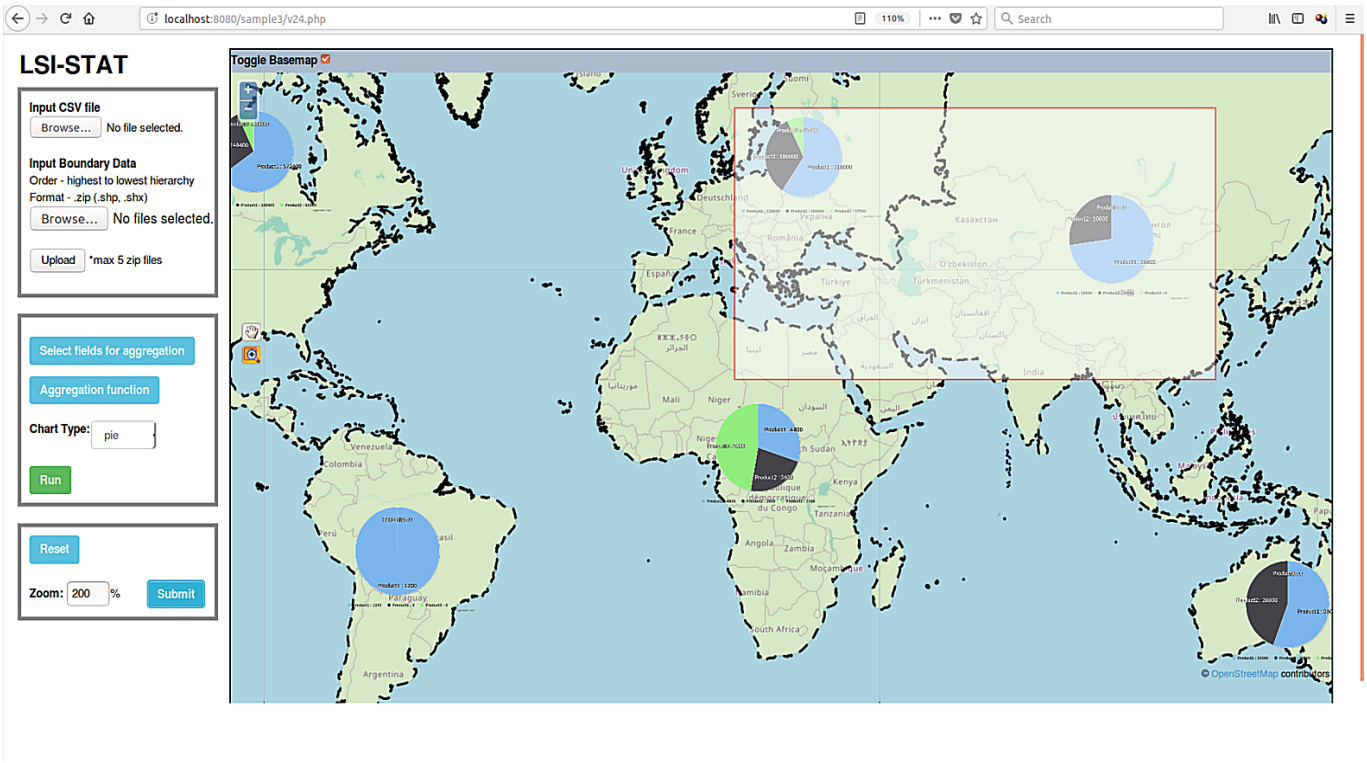

Figure 4.15: Draw Rectangle over area to be zoomed in

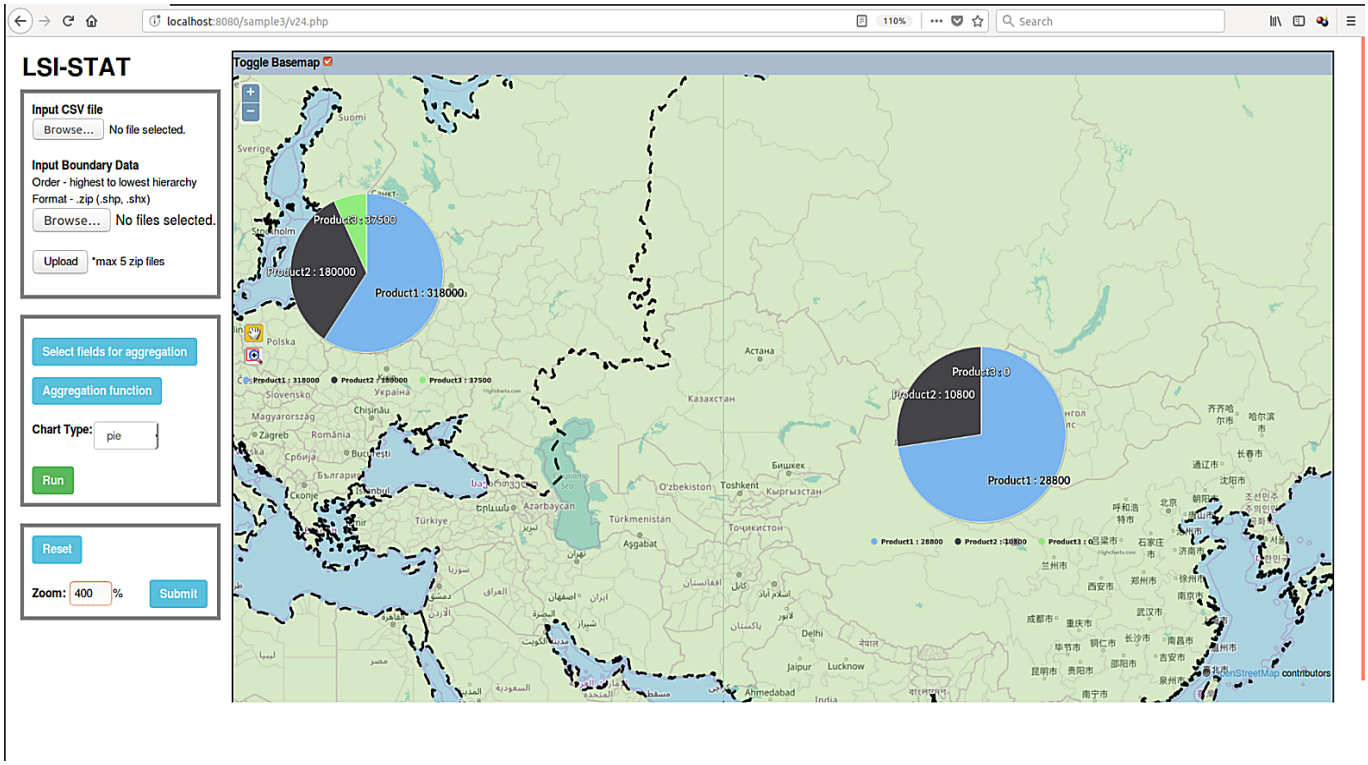

Figure 4.16: Zoomed into the selected area

Figure 4.15 shows rectangle drawn for zooming over the area and 4.16 shows the map zoomed into the selected area.

To zoom to a single or a set of charts, a rectangle can be selected around these charts on the map using this feature. This allows for a more extensive analysis and improves the user experience. Zoom-box controls were added to enable zooming in directly to a given extent by drawing a box on the map. Flowchart in figure A.11 shows the steps for providing zoom by area feature on map.

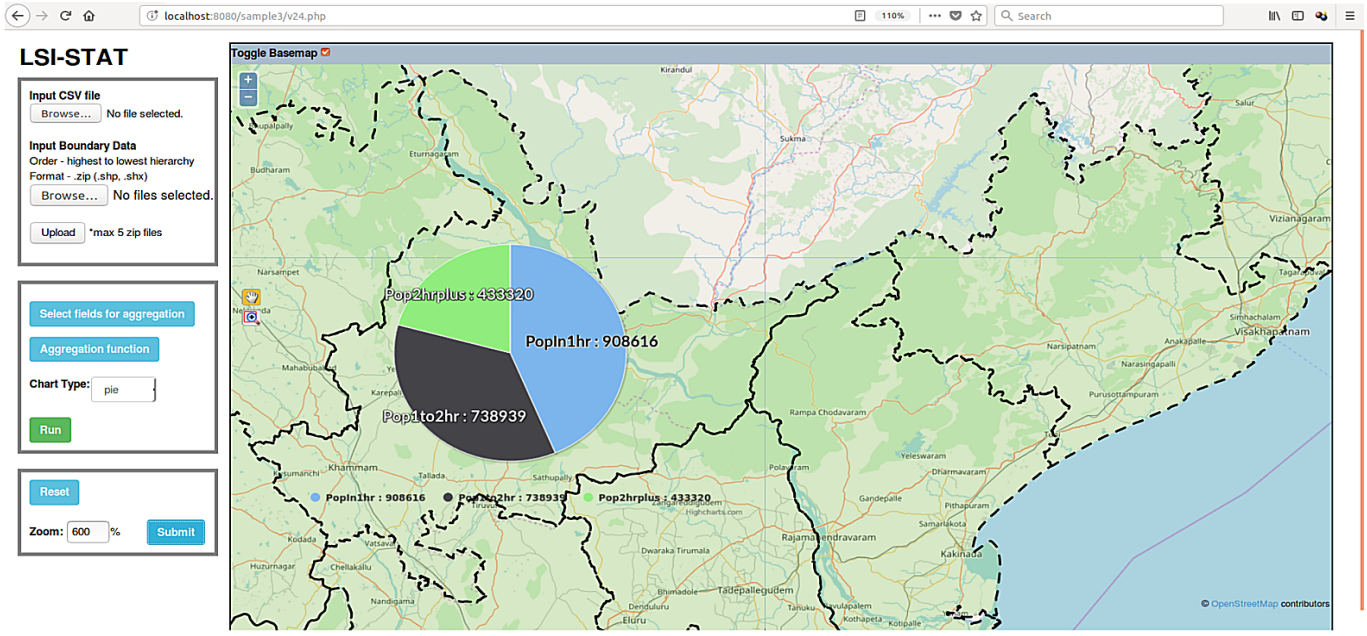

Figure 4.17: Initial Map

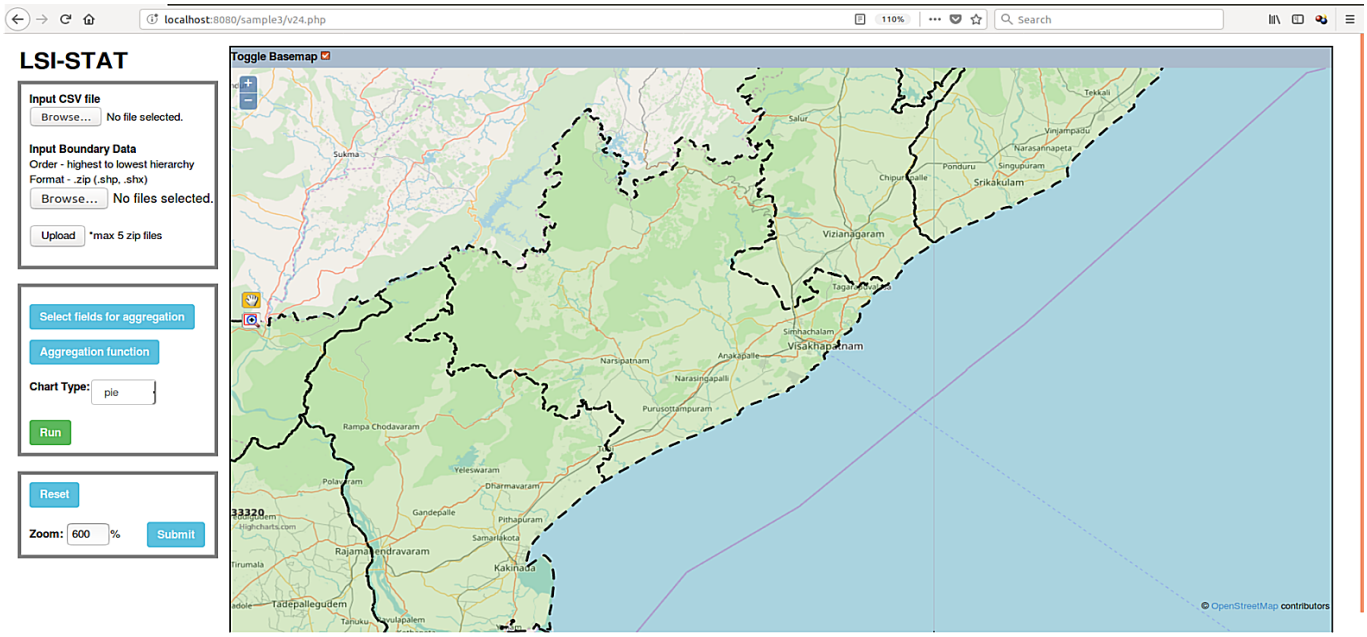

Figure 4.18: Right Shifted Map

Figure 4.17 shows the initial map and 4.18 shows the map after shifting it to the right.

Change in zoom level

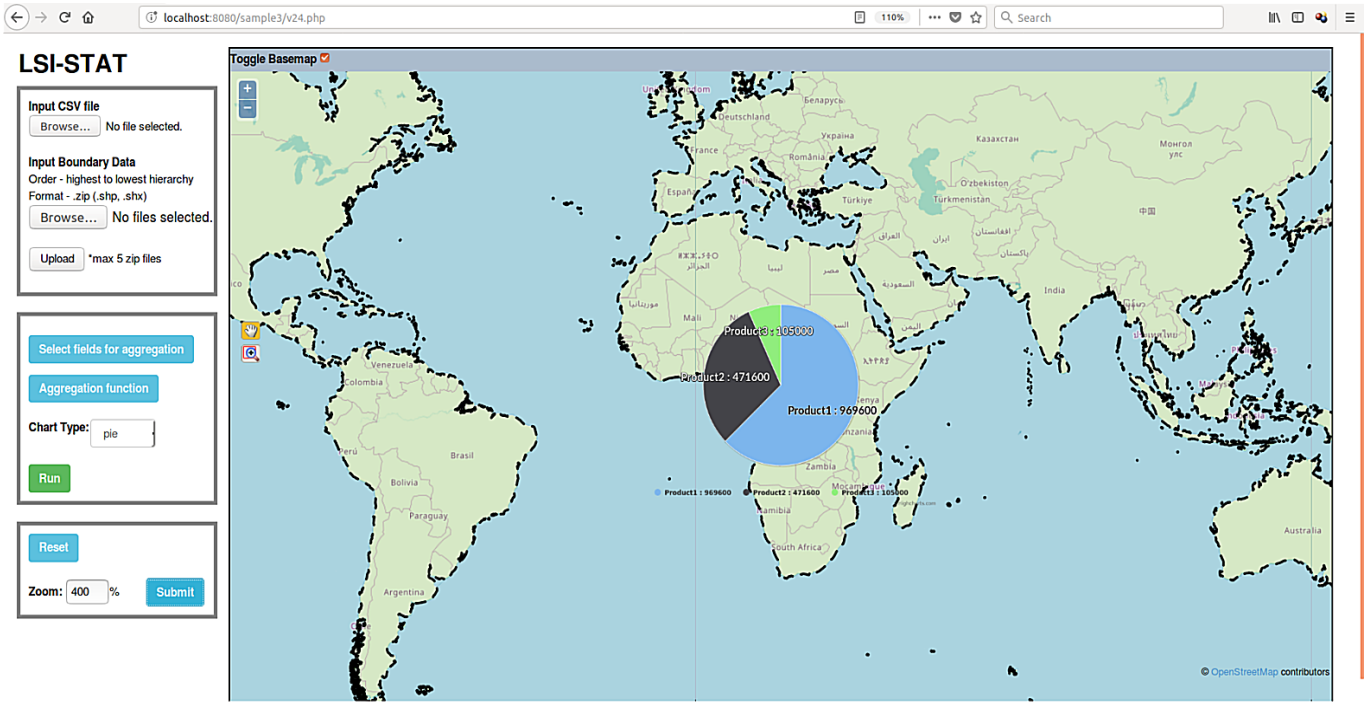

Figure 4.19: Initial Map with boundary file

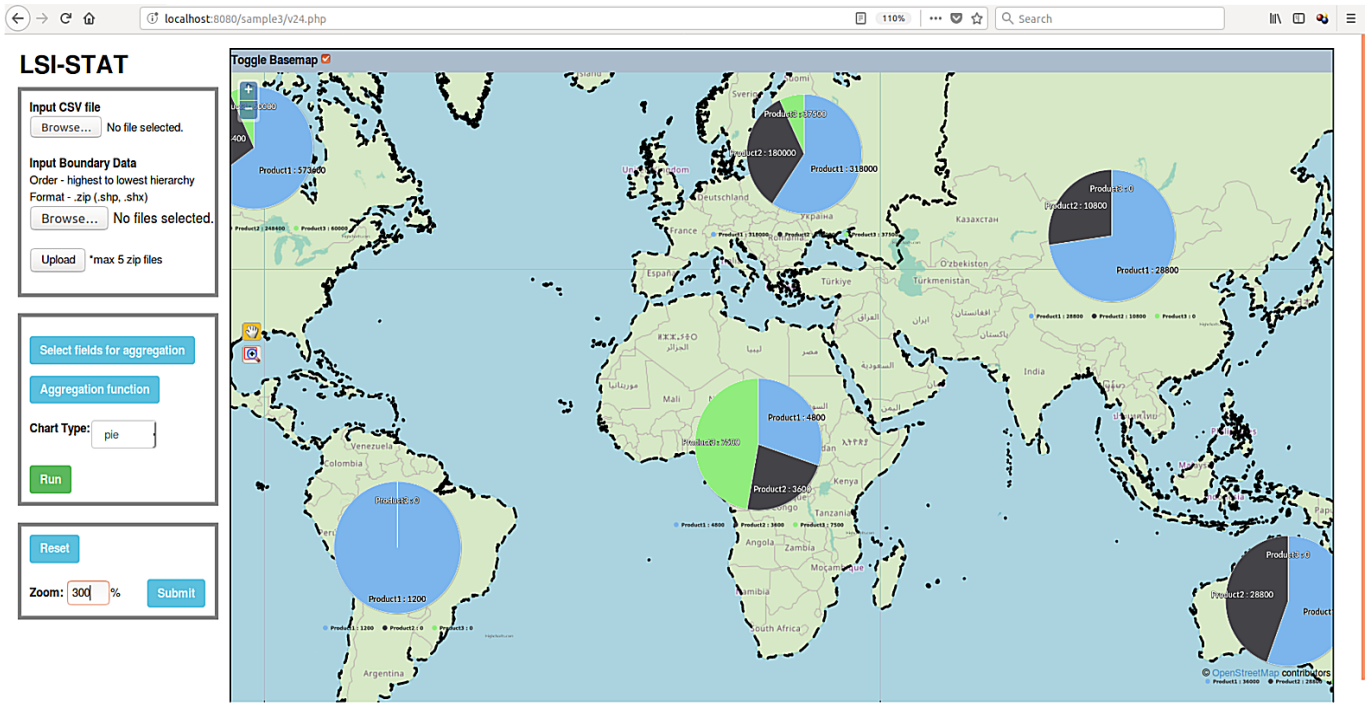

Figure 4.20: Map with change in boundary file on zooming in

Figure 4.19 shows the initial map with boundary file and 4.20 shows the map with change in boundary file on zooming into the map.

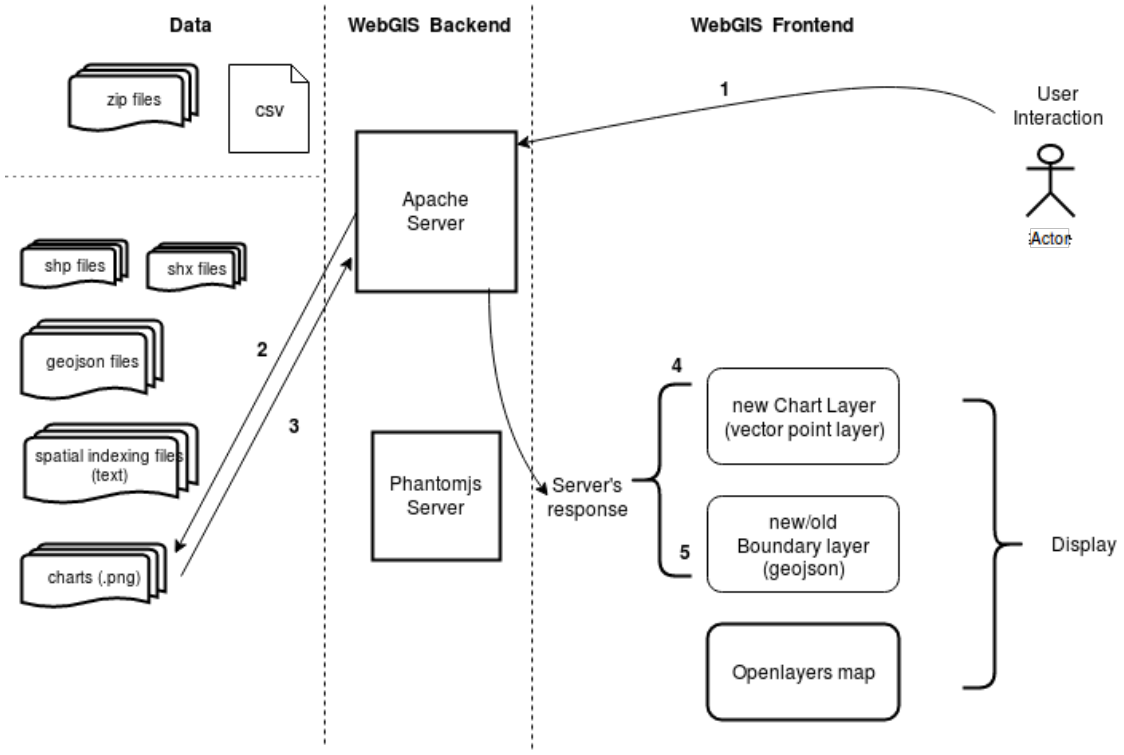

Figure 4.21: Change in zoom level

Figure 4.21 shows the working of the system on change in zoom level. Controls present on the map allow the user to zoom in or zoom out. New geo -hierarchy level is calculated based on the zoom level. If the new geo-hierarchy level is different from the previous one, new charts are to be displayed. These charts are obtained from chart cache by the tool. A new vector layer composed of another set of charts with recomputed chart size is created (4). This layer is then rendered along with the new geojson layer in the map panel of the browser (5). Otherwise, previous vector layer with recalculated chart size is rendered on the map.

The steps involved in data based processing have been given in flowcharts. Flowchart in figure A.12 shows the steps for creating a mapping between zoom level and boundary file to be displayed. Flowchart in figure A.13 shows the steps for fetching user selected fields. Flowchart in figure A.14 represents steps for setting chart options for chart generation. Flowchart in figure A.15 shows steps for initializing values before chart generation. Flowchart in figure A.16 represents steps for generating charts.

4.2.4 Data Visualization

4.2.4.1 Criteria for visualization

- Latitude and longitude values corresponding to a particular data point should be present in the user-given data file and it assumed to have correct data values. If data values are missing, then

that data point would be displayed on the map or the chart corresponding to the polygon in which it lies at a particular zoom level.

- If the centroid of the polygon lies inside the view area of the map, then the chart corresponding to that polygon is displayed. If not, then the particular chart is not displayed. The reason for this is that for any particular chart most of it should lie inside the view area for user to be able to understand the data point.

4.2.4.2 Generating charts for visualizing over map

There are two ways for generated charts to be visualized over a map. The generated charts can be sent to client and then client machine can visualize it over the map or this can be done by the server. Image transfer over the internet depends on the size of the image as well as the internet bandwidth. System performance may get affected if the internet bandwidth is low. Therefore, the server was chosen to add the charts on top of the map.

4.2.4.3 Visualizing charts over map

One of the approaches for visualizing charts over map is to create a new layer for each image generated and then load it on an OpenLayers map. However, generating many layers would take more time and overlaying them would be a challenge. Another approach would be to write your own SVG renderer for drawing charts using JavaScript on OpenLayers map. But this would involve writing JavaScript code for all chart types accurately. A better approach would be create a single vector layer and style each point with either SVG tag code for the required chart or directly add the chart image. This would save time, there would be no problem of overlay and would be accurate for different chart types as well. Using SVG tag code obtained from Highcharts SVG chart would involve storing SVG code for all charts. This would utilize more memory space. Hence, for displaying a generated chart on an OpenLayers map, a new layer was created composed of the chart images. This layer was added by the server to the OpenLayers map.

4.2.4.4 Zoom based computation

This allows user to explore the data interactively. With change in zoom level, the data represented on the map would change along with change in the number of charts displayed, size of charts, data values and boundary files (GeoJSON layer). First, the spatial aggregation was recomputed. Then, chart generation request were sent to the phantomjs server along with user-configured chart parameters. The exported charts were stored locally and added to map. In this way, more effective data visualizations are created which are easy to understand and interpret.

Boundary layers from higher to lower spatial hierarchy are shown as the user persistently drills down into the map. The boundary layer to be displayed is decided based on the current zoom level of the

map. The chart displayed should have minimum dimensions to be clearly visible to the end-user. Only if the size of the smallest chart displayed is greater than the minimum dimensions of a chart required for clarity, then boundary layer lower in hierarchy is shown in map. The rationale behind this is that a layer lower in spatial hierarchy will definitely have polygons smaller than a layer higher and the chart that fits inside a smaller polygon would be smaller in size. Until every chart from that layer is clearly visible to the user, there is no use of displaying charts smaller than the current size to the user. The whole point of the tool is to help the user visualize his data with the help of charts interactively on the map. Without proper clarity in charts, user will not be able to analyze data or gain any significant insights.

Flowchart in figure A.17 show steps for overlaying charts on map. Flowchart in figure A.18 show the steps followed on change in current zoom level.

Chapter 5

GeoBi - Online retail case study

TO show the utility of LSI-STAT we worked on a dataset that was publicly available to understand how the tool can infer this data and present results. This case study is taken as a usecase for showing the utility of this tool.In order to grow this business, an online retailer needs to understand his customers across regions and find answers for questions such as - what their popular products are, regions where there is a demand for these, regional preferences and many others. This helps him in catching the straying customers and giving incentives, resonating with your customers, improving customer service, helping in setting product prices, deciding on shop location and planning promotional offers [51].

5.1 Input Data Used

Online Retail data The online retail dataset had details of three products sold in different countries in January 2009 [52].

Data characteristics:

- Time-period: 1st January, 2009 to 31st January, 2009 (day-wise sales data)
- Data: 1000 records
- Type: Spatio-temporal Dataset
- Attributes (30): Transaction_Date, Product, Price, Payment_Type, Name, City. State, Country, Account_Created, Last_Login, Transaction_date.year, Transaction_date.month, Transaction_date.day-of-month, Transaction_date.day-of-week, Transaction_date.hour, Transaction_date.minute, Account_Created.year, Account_Created.month, Account_Created.day-of-month, Account_Created.day-of-week, Account_Created.hour, Account_Created.minute, Last_Login.year, Last_Login.month, Last_Login.day-of-month, Last_Login.day-of-week, Last_Login.hour, Last_Login.minute
- Geo-hierarchy: 4 levels (city, state, country and continent)
- Temporal-hierarchy: 2 levels (day, month)

Boundary shapefiles Along with this, boundary shapefiles of World, Continent, UK and Canada were used [53, 54, 55].

Preparing input files for visualization and analysis using tool Location data - latitude and longitude obtained from google maps were added to the csv input data file for geo-visualization. Attributes present in the data input file were found to be insufficient. Hence, additional attributes were added for testing different hypothesis. These were:

- Payment_Type_Visa, Payment_Type_Amex, Payment_Type_MasterCard, Payment_Type_Diner
- Product1Sales, Product2Sales, Product3Sales
- Hourly Total Sales

Publicly available data has been used in this case study whose quality is not known and it is assumed that the information available is useful for the study. The visualizations generated by the tool are only for demonstrating its utility and application. The results obtained should not be used for making inferences and deciding policies for the region. Also, some of the values were modified for better illustrations. The charts generated were code-checked for any discrepancy against the actual values obtained from the dataset.

5.2 Finding patterns through data visualization and exploration

The modified input data was added to the tool for visualization. The outcomes obtained after doing an analysis of sales of all products, payment options used by the customer and hour-wise products purchased would hugely benefit an online retailer. Such analysis involves asking many different questions.

- How many products were sold all over the world?
- How many products were sold in each continent?
- How much quantity of each product was sold all over the world?
- How much quantity of each product was sold in each continent?
- How many products were sold in United Kingdom (UK) every hour on a particular day?
- How many products were sold in different provinces of Canada at 10am and entire day?
- How many products were sold in different provinces of Canada throughout the day?

The results of our analysis for the above questions on the dataset are presented in the next section. Multiple visualizations can be generated by the platform using combinations of different attributes, time-periods and geo-locations. Attributes displayed on charts could be single, multiple or aggregated.

Similarly, time-period shown could be single, multiple or aggregated. The geo-location could be a single point or multiple points. In this case study, we show the results obtained on generating visualizations for some of these combinations of attribute, time-period and geo-location.

5.3 Case Study Results

In order to demonstrate that visualizations can reveal patterns and trends easily compared to only statistical analysis, tables corresponding to each of them have been shown along with the questions can be answered easily through the visualization generated by the platform.

5.3.1 Aggregated Attributes, Aggregated over Time-period, Single Geo-location

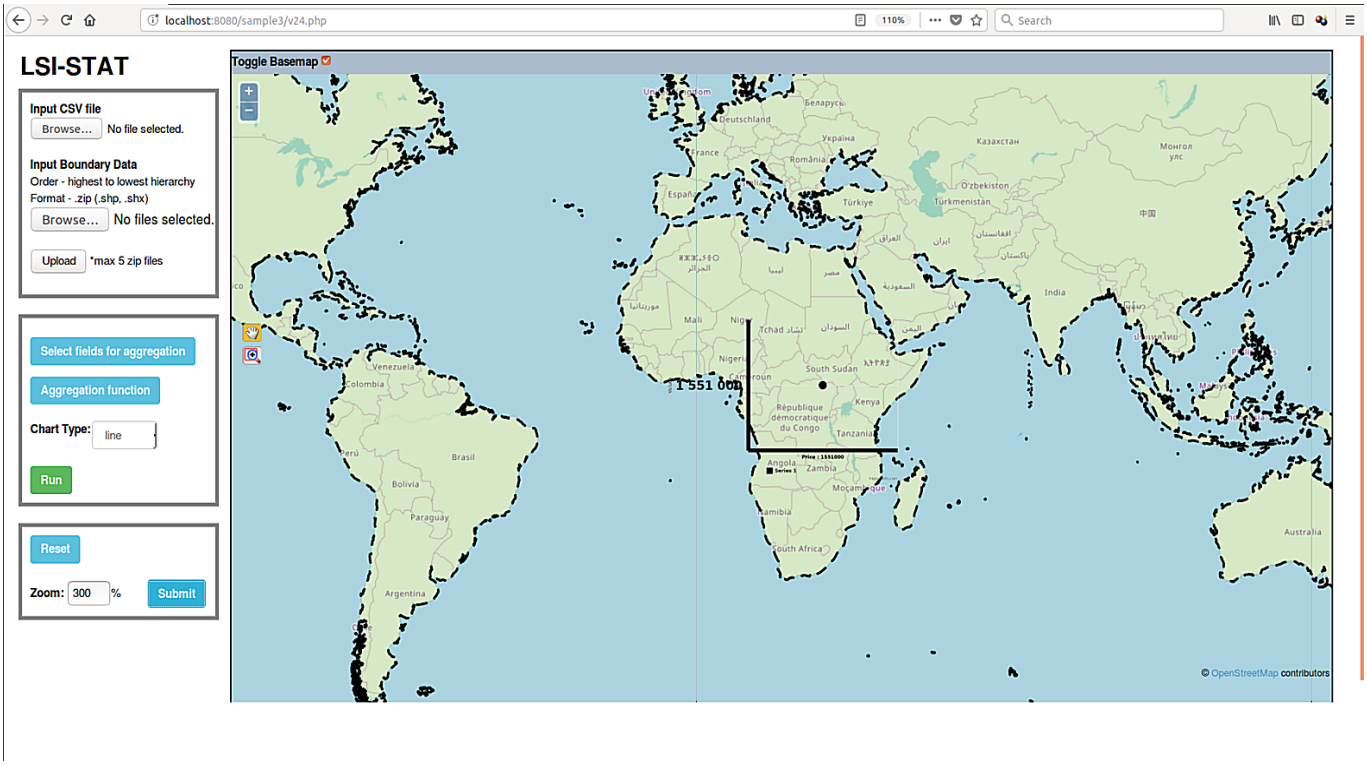

Figure 5.1: Figure shows total products sold all over the world

Figure 5.1 shows the quantity of each product sold all over the world. The visualization helps us answer the following question.

How many products were sold all over the world?

The total sales for January 2009 all over the world was found to be 15,51,000. Figure above shows the same.

5.3.2 Aggregated Attributes, Aggregated over Time-period, Multiple Geo-locations

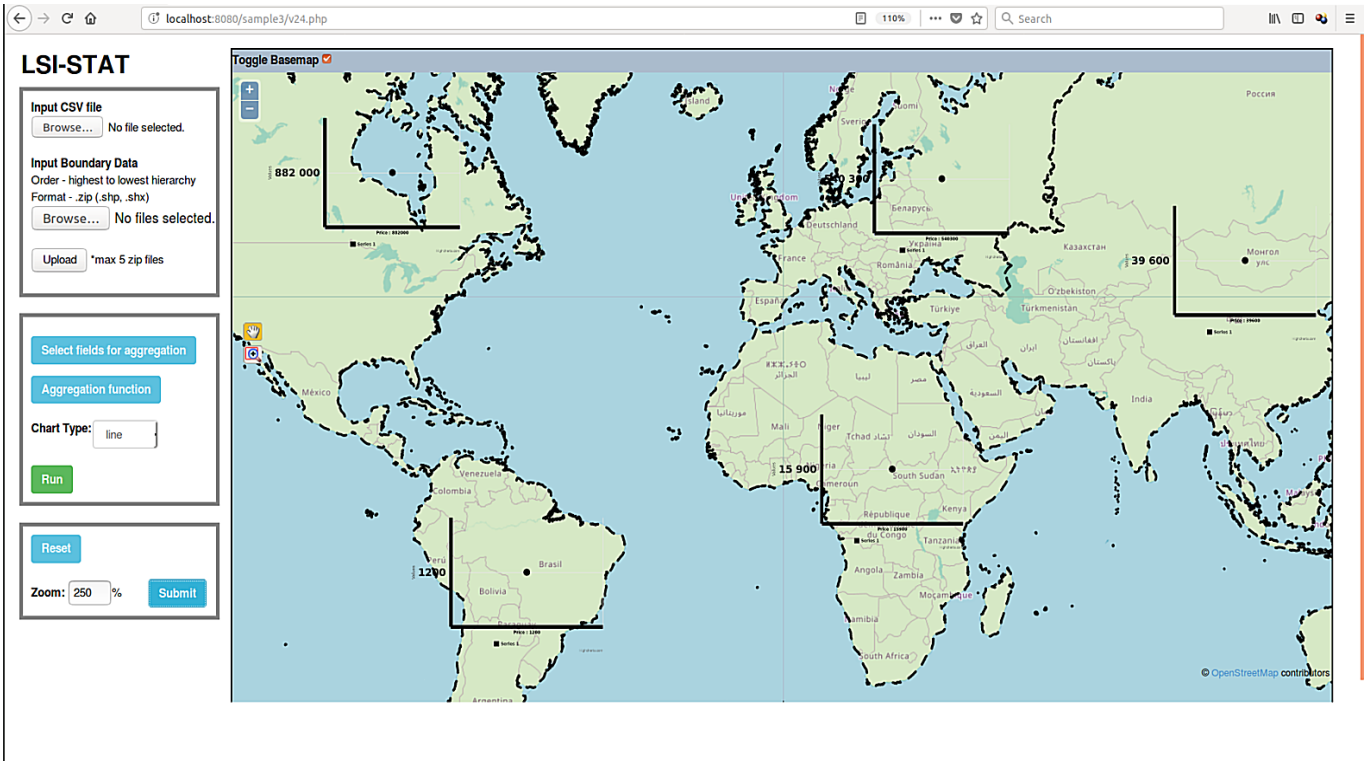

Figure 5.2: Figure shows total products sold in each continent

A drill-down into map shows the sales of each continent for the January 2009 (Figure 5.2). The visualization helps us answer the following questions.

In which continent were maximum products sold?

North America (882000)

In which continent were minimum products sold?

South America (1200)

What is the order of continents from maximum products sold to minimum products sold?

North America (8,82,000) followed by Europe (5,40,300), Oceania (72,000), Asia (39,600), Africa (15,900) and South America (1200).

Which continents constitute together constitute large market shares?

North America and Europe together constitute more than 95% of the total market share.

Continent	Total Sales
North America	882000
Oceania	72000
Africa	15900
Europe	540300
Asia	39600
South America	1200
Australia	64800

Table 5.1: Total Sales in each continent

5.3.3 Multiple Attributes, Aggregated over Time-period, Single Geo-location

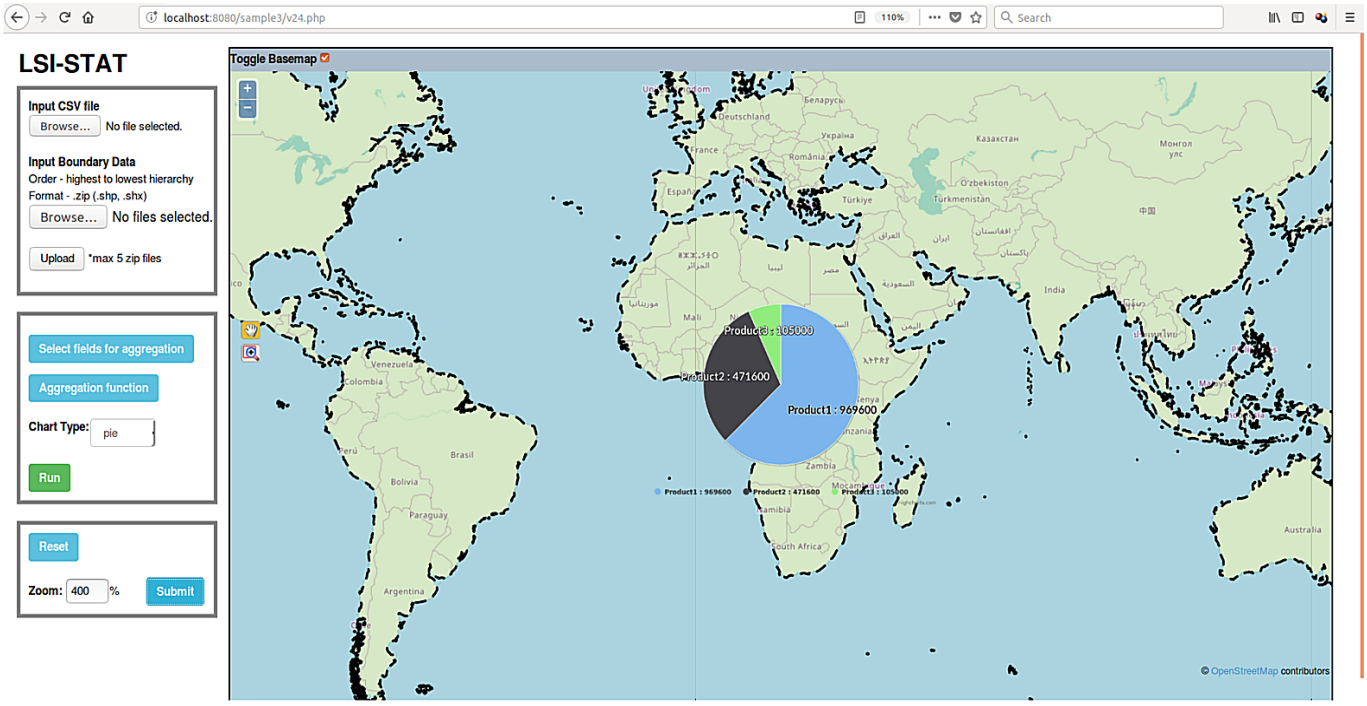

Figure 5.3: Figure shows the quantity of each product sold all over the world

Figure 5.3 shows the quantity of each product sold all over the world. The visualization helps us answer the following question.

What is the order of products from most popular to least popular?

Product 1 (969600) was found to be the most popular product, followed by Product2 (471600) and Product3 (105000).

Product	Total Sales
Product1	969600
Product2	471600
Product3	105000

Table 5.2: Quantity of each product all over the world

5.3.4 Multiple Attributes, Aggregated over Time-period, Multiple Geo-locations

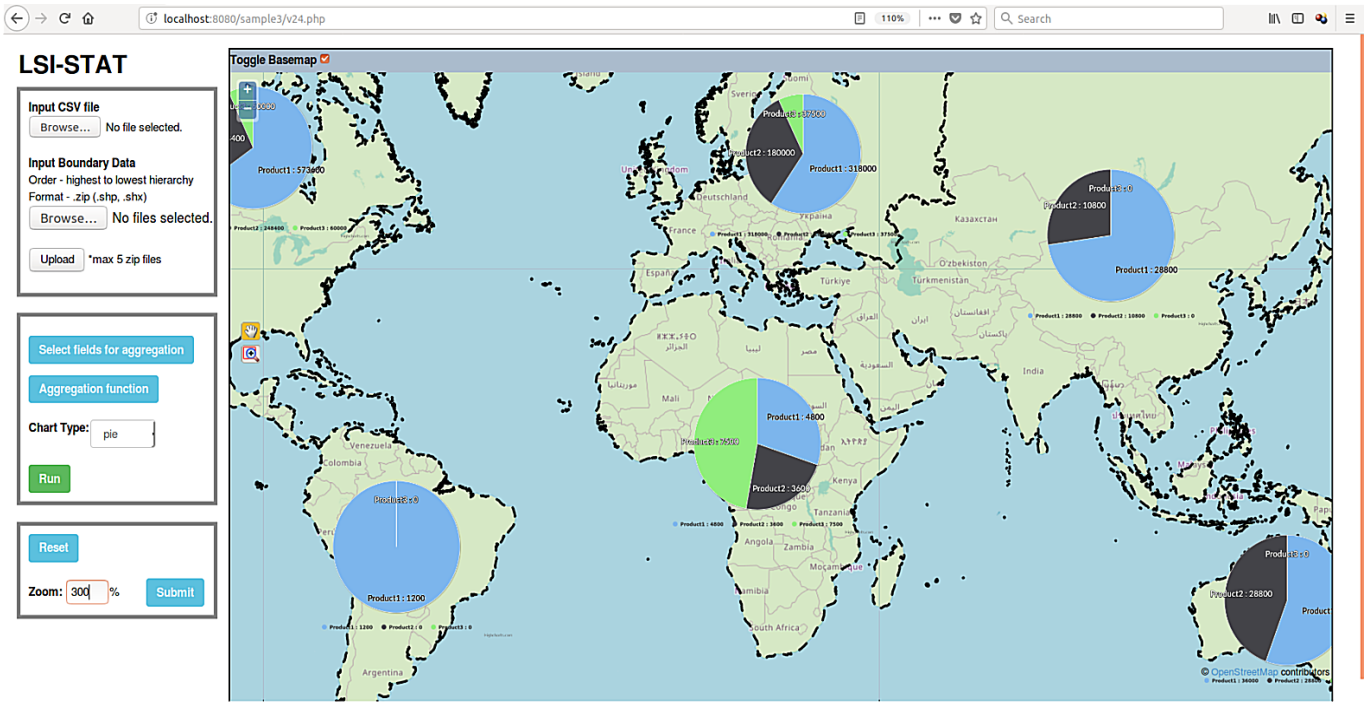

Figure 5.4: Figure shows the quantity of each product sold in each continent

A further drill down into the map shows the sales distribution for all products for each continent (Figure 5.4). The visualization helps us answer the following questions.

In which continents, only Product1 was sold?

South America and Oceania

In which continents, only Product1 and Product2 were sold?

Asia and Australia

In which continents, all the three products were sold?

Africa, North America and Europe

What is the order of popularity for the continent North America?

Product1, Product2 and then Product3

In which all continents, is the order of popularity - Product1, Product2 and then Product3?

All continents except Africa

What is the order of popularity for the continent Africa?

Product3, Product1 and then Product2

Continent	Product1 Sales	Product2 Sales	Product3 Sales
North America	573600	248400	60000
Oceania	7200	0	0
Africa	4800	3600	7500
Europe	318000	180000	37500
Asia	28800	10800	0
South America	1200	0	0
Australia	36000	28800	0

Table 5.3: Quantity of each product in each continent

5.3.5 Multiple Attributes, Multiple Time-periods, Single Geo-location

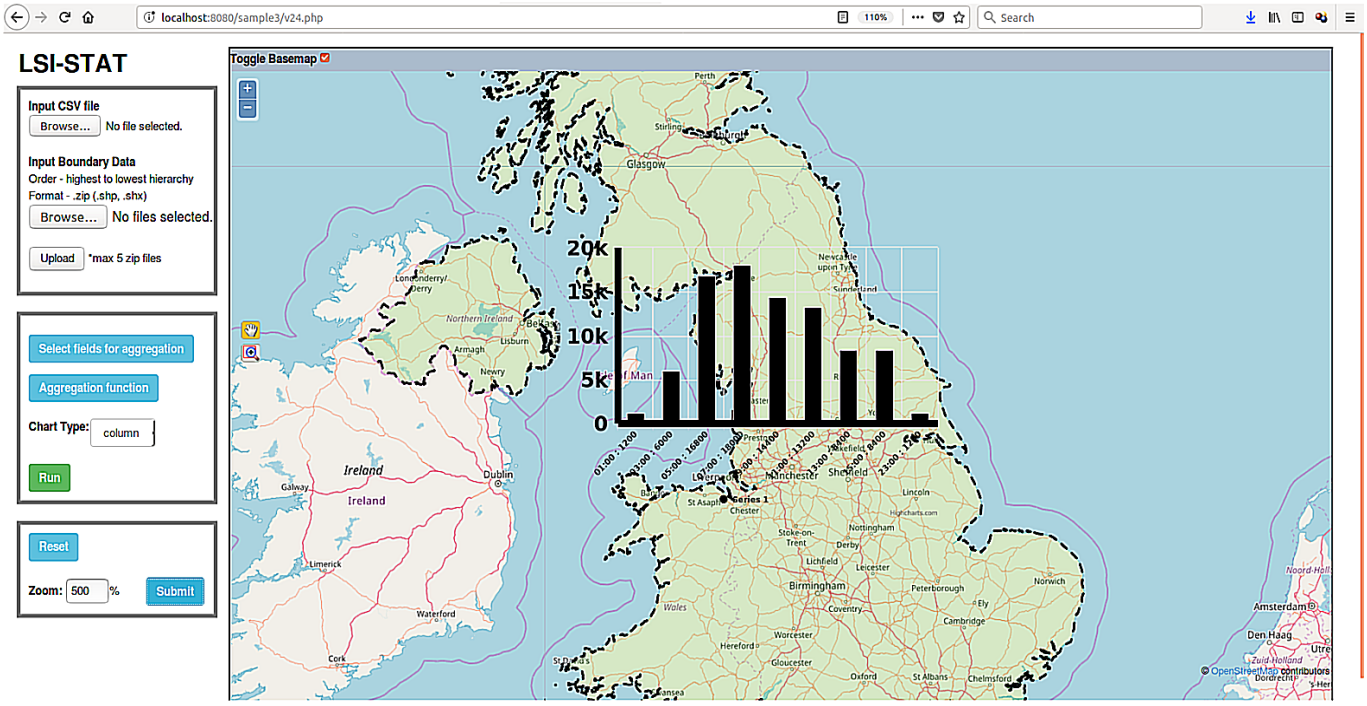

Figure 5.5: Figure shows the total number of products sold in United Kingdom (UK) every hour on a particular day

Figure 5.5 shows the total number of products sold in United Kingdom (UK) every hour on a particular day. The visualization helps us answer the following questions.

At what time did maximum sales occur and what was the quantity?

At 7am, 18000

During which time-period were a large number of products sold?

Between 7am to 11am

What was the trend observed?

Sales increased gradually from 1am onwards, a peak was observed at around 7am and then it began to decrease till 3pm.

From which hour did sales decline?

7am

Did sales halt during any time-period?

Sales almost came to a halt around 3pm.

Hour	Product Sales
1:00	1200
3:00	6000
5:00	16800
7:00	18000
9:00	14400
11:00	13200
13:00	8400
15:00	8400
23:00	1200

Table 5.4: Total products sold in United Kingdom every hour

5.3.6 Multiple Attributes, Single Time-period, Multiple Geo-locations

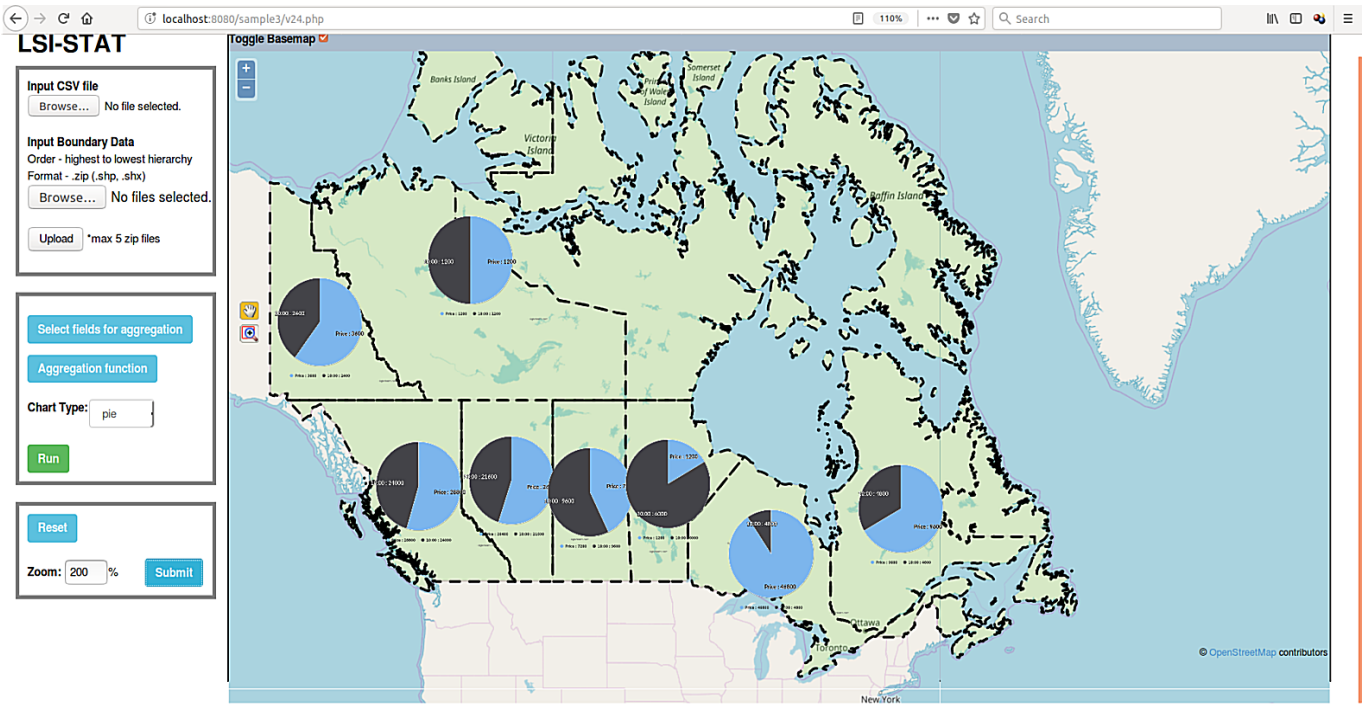

Figure 5.6: Figure shows the total number of products sold in different provinces of Canada at 10am and entire day

Figure 5.6 shows the total number of products sold in different provinces of Canada at 10am and entire day. The visualization helps us answer the following questions.

Was the percentage share of sales at 10am more than 50% or less?

It was less than 50% in all of them

In which provinces was the share the least and the most of all the provinces?

Least was in North Western Territories and Maximum in British Columbia

Province	10:00	Total Sales
Alberta	10800	144000
British Columbia	14400	115200
Manitoba	6000	87600
Northwest Territories	1200	100800
Ontario	3600	68400
Quebec	4800	111600
Alberta	10800	99600
British Columbia	9600	106800
Ontario	1200	80400
Saskatchewan	9600	92400
Yukon Territory	2400	58800

Table 5.5: Total number of products sold in different provinces of Canada at 10am and entire day

5.3.7 Multiple Attributes, Multiple Time-periods, Multiple Geo-locations

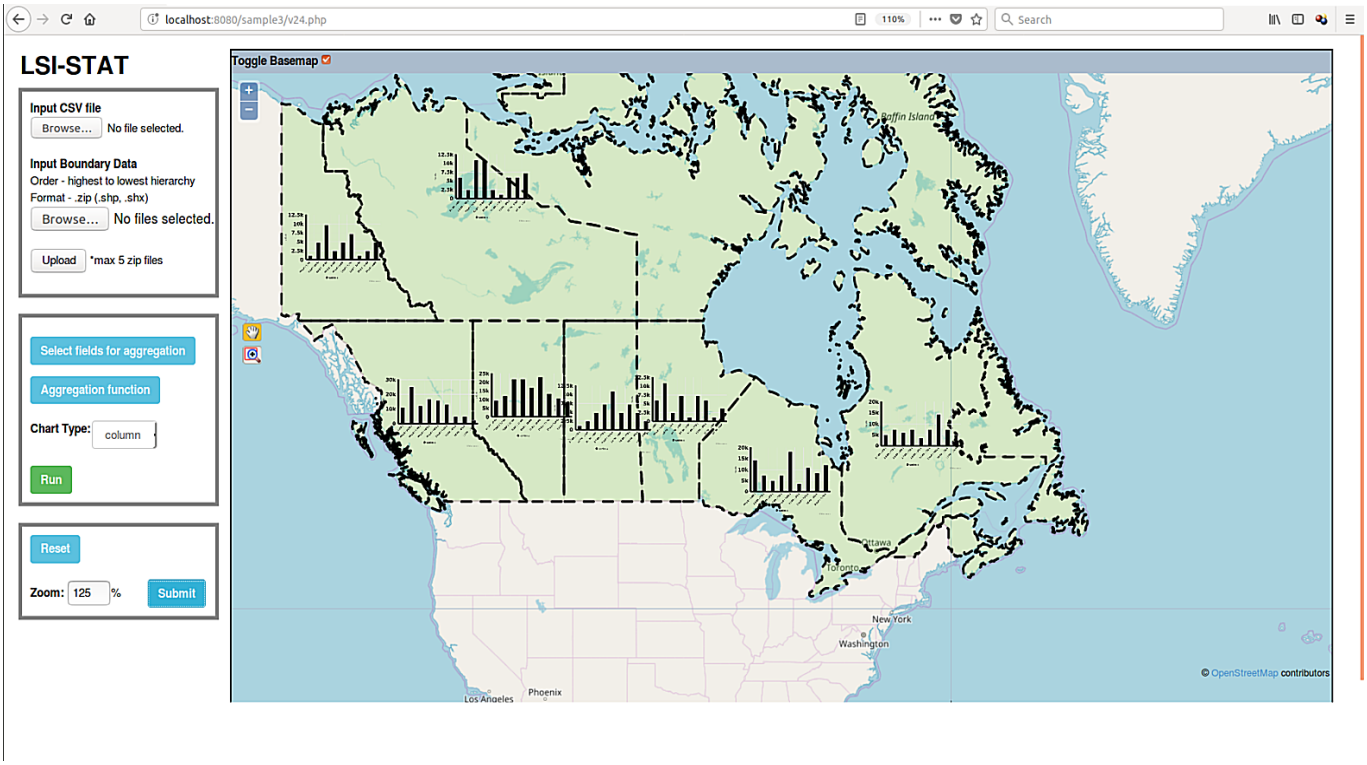

Figure 5.7: Figure shows the total number of products sold in different provinces of Canada in the entire day

Figure 5.7 shows the total number of products sold in different provinces of Canada in the entire day. The visualization helps us answer the following questions.

Is there any general trend or pattern followed across all the provinces?

No

What is the general trend or pattern followed in Alberta province?

The number of products sold increases continuously to 18,000 at 7am and then decreases continuously. The number of products sold at 11pm when the day ends is the same as the number of products sold when sales began at 1am.

Do any of the neighboring provinces have peak sales at 7am as Alberta?

No

Which of the provinces has the provinces has the highest sales in the entire day?

Alberta has highest sales at 7am (18000)

How many of the provinces have peak sales for continuously for two or more hours of the day?

Northwest Territories has peak sales continuously from 5am to 7am.

Province	01:00	03:00	05:00	07:00	09:00	11:00	13:00	15:00	23:00
Alberta	1200	6000	16800	18000	14400	13200	8400	8400	1200
British Columbia	7200	10800	4800	10800	6000	6000	2400	1200	1200
Manitoba	6000	10800	2400	7200	1200	7200	6000	1200	3600
Northwest Territories	6000	2400	10800	10800	2400	1200	6000	6000	7200
Ontario	3600	6000	2400	3600	10800	1200	7200	1200	2400
Quebec	4800	7200	6000	7200	3600	7200	14400	7200	2400
Alberta	8400	6000	4800	3600	2400	9600	4800	2400	7200
British Columbia	3600	14400	7200	6000	9600	7200	2400	3600	1200
Ontario	10800	1200	2400	3600	7200	2400	3600	7200	9600
Saskatchewan	1200	2400	4800	7200	10800	4800	7200	4800	2400
Yukon Territory	1200	4800	9600	2400	4800	7200	1200	2400	4800

Table 5.6: Total number of products sold in different provinces of Canada in the entire day

Chapter 6

Visualizing Health Services access and reach in Khammam district, Telangana case study

To demonstrate the utility and usability of this tool, case study on visualizing health services has been presented here. This can help policymakers find inaccessible places, places which have limited access to healthcare facilities, lack certain healthcare services or have fewer number of healthcare facilities. This helps them easily locate places to setup new healthcare facilities and identify services required to be added to a healthcare facility.

6.1 Aarogyasri Rajiv Health Insurance Scheme, Telangana

It is a unique scheme started by the government of Telangana for the benefit of the families living below poverty line by providing them financial protection upto Rs 2 Lakh per year for serious ailments involving surgery or hospitalization. The program provisions access to health care and services for the poor. 938 treatments have been covered under this scheme.

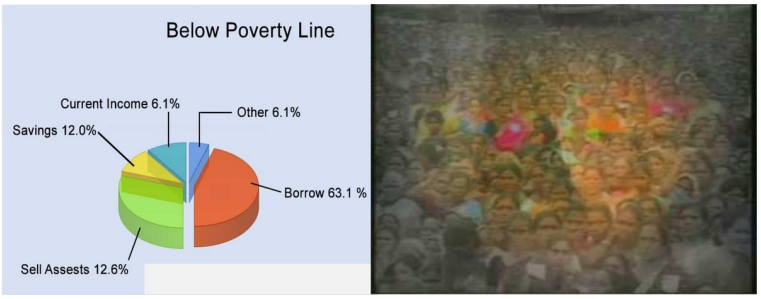

Figure 6.1: People below the poverty line

It was realized by the government that due to huge increase in healthcare costs, it is leading to rural indebtness and large number of people living below the poverty line have to borrow or sell assets to pay for hospitalization. Figure 6.1 shows how a person staying below the poverty line spends his income [56].

This scheme was launched on 9th June 2006 and has been in operation ever since. A network of 344 hospitals has been setup for this across several districts in Telangana. This includes Primary Healthcare Centers (PHC)s, Government and Private hospitals. The main features of the scheme as per 2009 statistics are given below [57].

- Health insurance for 2.03 crores BPL Families
- 942 Procedures
- 344 Network Hospitals
- 1100 Surgeries done everyday costing 3.5cr per day
- Follow-up treatment for 1 year for 121 procedures
- Over 3.3 lakh surgeries done so far.
- At a cost of 1032 cr
- Contribution of Government Hospitals in about 15
- Rs 1334 cr budget estimated for 2009-10
- Rs 925 cr from budget and 409 from CMRF
- Aarogyasri Trust runs with 57 trust staff
- All Trauma Cases covered
- Cochlear implantation for children upto 12 yrs included
- 3047 Aarogyamitras in Network hospital/ PHC for Patient care
- 3,44,476 surgeries/therapies done so far

The scheme provides coverage under the following surgery categories: General Surgery, ENT, Opthamology, Orthopaedics, Surgical Gastroenterology, Cardio Thorasic surgery, Pediatric Surgery, Genito Urinary surgery, Neuro surgery, Surgical Oncology, Medical oncology, Radio Oncology, Plastic Surgery, Polytrauma, Prostheses, Critical care, General Medicine, Infectious Diseases, Paediatric Intensive Care, Neonatal Intensive care, Paediatric General, Cardiology, Nephrology, Neurology, Pulmonology, Dermatalogy, Rheumatology, Endocrinology, Gastroenterology, Cochlear Implantation [58].

To get an initial understanding of the data, the total number of cases that occured and the amount recovered in different months for the years 2014 and 2015 were plotted in figures 6.2 and 6.3 respectively.

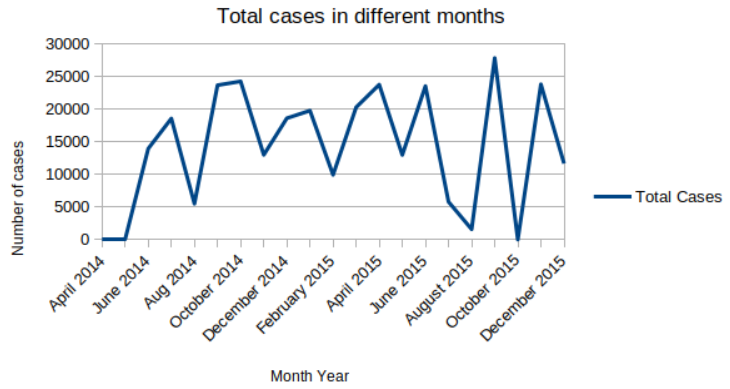

Figure 6.2: Total cases in different months

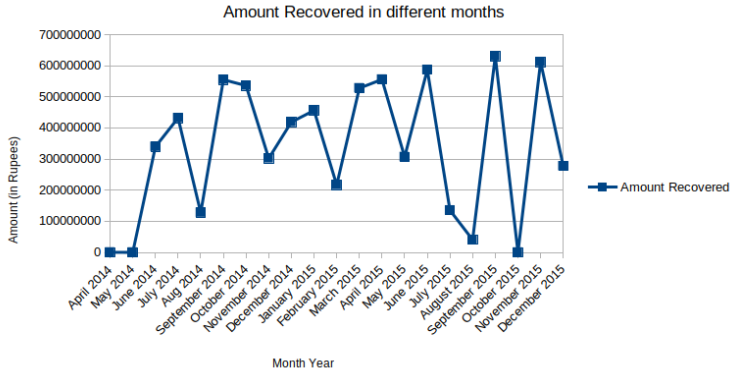

Figure 6.3: Amount Recovered in different months

It can be seen from the graph in figures 6.2 and 6.3 that the number of cases and the amount recovered have not been uniform over the time-period. This fluctuation in total cases in Khammam could be due to effect of change in seasons, epidemic in one or more regions, increase in awareness of the scheme or modifications in the scheme policy. This leads us to further explore this health dataset with the help of LSI-STAT platform.

6.2 Study Area

Khammam district located in eastern region of Telangana state in India was selected for this case study. Figure 6.4 shows Khammam district on world map [59] and figure 6.5 represents Khammam district map [60]. It has a total population of 1,401,639 and a density of 320 per square km [61]. It is subdivided into mandals. The city of Khammam is the headquarters of the district.

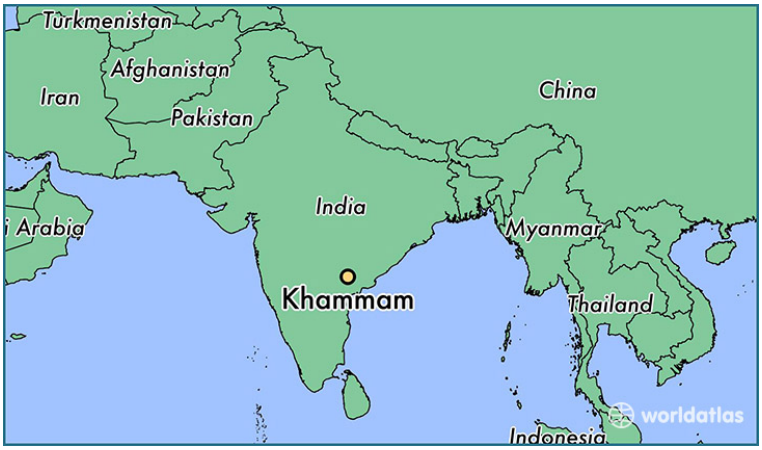

Figure 6.4: Locating Khammam on world map

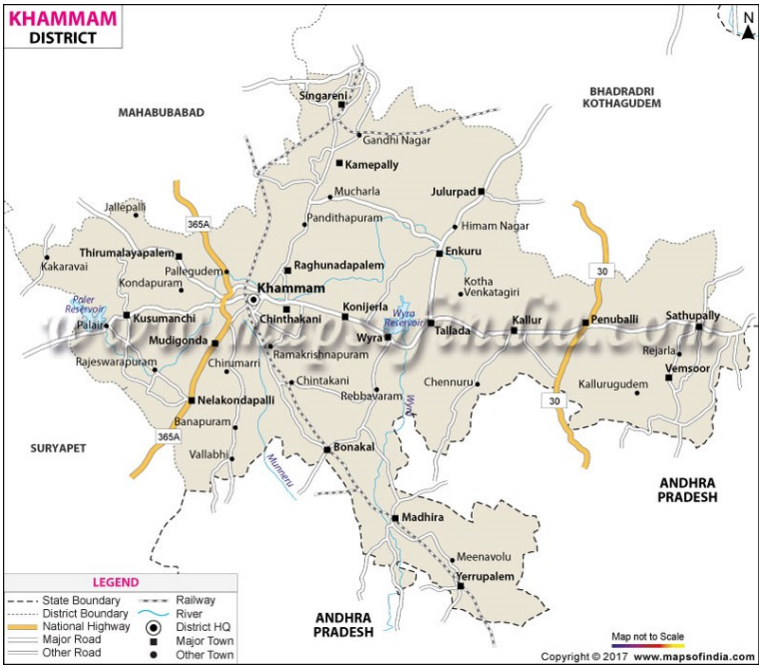

Figure 6.5: Khammam district map

Inorder to remove cases of in-migration to Khammam and out-migration from Khammam to other districts for medical treatment, records of patients residing in Khammam district and availing treatments in hospitals located in Khammam were used in this analysis. By doing so, Khammam district acts as a complete system. For this analysis, It is assumed that the population in a mandal was uniformly distributed across its area.

6.3 Data Used

Input datasets:

- Patients claims data of Rajiv Aarogyasri Health Insurance Scheme in Khammam district of Telangana
- Census data of Khammam district
- Travel time estimates for a patient to reach the nearest hospital
- Administrative level boundary shapefiles of India.

Patients claims data of Rajiv Aarogyasri Health Insurance Scheme (Khammam district of Telangana)
Dataset characteristics:

- Data: More than 6000 records.
- Type: Spatio-temporal Dataset
- Time-period: July 2013 to July 2015 (Day-wise counts for 24 months).
- Data Attributes (27): Card_No, CardNo_MID, Age, Sex, Caste, Category_Code , Category_Name, Surgery_Code, Surgery_Name, Village, Mandal_Name, District, Lat, Long, Preauth_Date, Preauth_Amount, Claim_Date, Claim_Amount, Hospital_Name, Hospital_Type, Hospital_Location, Hospital_District, Surgery_Date, Discharge_Date, Mortality_Yes/No, Mortality_Date, Source Registration
- Geographical-hierarchy: 3 levels (Village, Mandal, District)
- Temporal-hierarchy: 3 levels (Day, Month, Year)

Census data of Khammam district
Population and area of mandals for year 2011 required for analysis were obtained from District Handbook of Andhra Pradesh.

Processed dataset:
Travel time estimates for a patient to reach the nearest hospital
This dataset was generated since it was required for answering queries for health facility analysis. For example, it is necessary to find the hospitals which are accessible to a patient within one hour. The time taken for a patient to travel to a hospital needs to be computed for answering such questions.

To calculate this, road travel time was used in our analysis. 80% of passenger traffic is known to be carried by roads according to National Highways Authority of India. Therefore, the time was calculated based on google maps data since it also accounts for traffic along the way. This approach for calculating time taken was taken in as close to real-life scenario as possible. Figure 6.6 represents the road network of Khammam district.

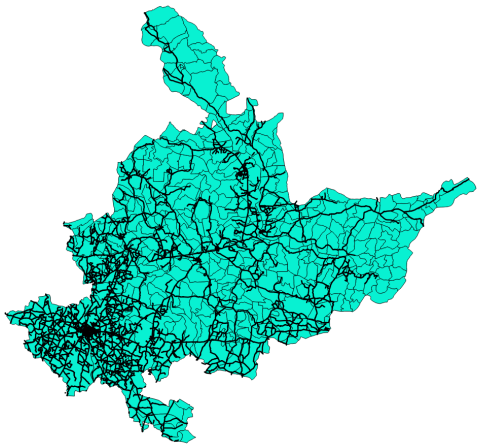

Figure 6.6: Road network of Khammam district

Instead of calculating time taken, distance can also be used as a measure for calculating proximity to a source such as calculating Euclidean, Manhattan or Minkowski distances between source and destination. However, it has been found that time taken is a more accurate measure than distance for measuring accessibility [63].

Eleven hospitals which have been included in the insurance scheme since the year 2013 and were considered in this study are as follows:

1. Area Hospital - Badrachalam
2. Area Hospital Kothagudem
3. Area Hospital Penuballi
4. Area Hospital - Sathupally
5. Area Hospital Paloncha
6. District Hospital-Khammam
7. Mamatha General And Super S pecialty Hospital
8. Jayabharathi Multispecialty Hospital
9. SriRaksha Hospitals
10. Sri Ram Kidney Infertility and Laparoscopic center
11. Srujan Ortho and Accident Care Hospital

Preparing input files for visualization and analysis using tool Attributes from the input datasets - Census data of khammam district and travel time estimates for a patient to reach the nearest hospital were added to the data file along with latitude and longitude data of all hospitals. The latitude and longitude data was obtained from google maps.

Additional attributes added to the data file: Population, Latitude, Longitude, Time taken, Area, Population Density, Distance, Pop_1hr (Population with travel distance of 1 hour), Pop_1to2hr, Pop_2hrplus, Areain1hr, Area1to2hr, Area2hrplus.

Publicly available data has been used in this case study whose quality is not known and it is assumed that the information available is useful for the study. The visualizations generated by the tool are only for demonstrating its utility and application. The results obtained should not be used for making inferences and deciding policies for the region. Also, some of the values were modified for better illustrations. The charts generated were code-checked for any discrepancy against the actual values obtained from the dataset.

6.4 Spatio-temporal Patterns evident from the data

Data exploration is a very important step of data analysis. It helps the user to understand the overall data characteristics and identify interesting variables to construct a hypothesis. For exploring the datasets, different visualizations were generated.

Inorder to find patterns in the dataset, we further analyze health facility. There are three main criteria for health facility analysis - provision of services, reach and accessibility. The health framework consists of three important dimensions - accessibility, provision of services (availability) and reach.

- Accessibility here refers to access to health care. Are the healthcare providers available when you need them? Do they give appointments when required?
- Availability refers to the opportunity to access the right type of health care services when needed as well as having the appropriate type of service providers, materials, and equipment.
- Reach refers to the physical distance or travel time between the service delivery point (primary or secondary health facility) and the user. Are the roads well connected between the user and the health facility? What are the alternative modes of transport in the region?

Health data visualization

From the dataset it was found that Polytrauma is the category in which most patients avail treatment under the scheme. About 45.06% of the cases are polytrauma cases. So, with the help of this platform, mandals in which these cases occur were found. Figure 6.7 is the road dataset of Khammam which helps us understand if these locations are well connected by roads so that the patients can reach hospitals in time.

Cardiac is one of the crucial health categories because in this the patient requires urgent medical attention. It is well known that cardiac service should be available to a patient within the golden hour. Also, patients are more likely to enroll in cardiac rehabilitation (CR) program if their travel time is less than sixty minutes to the nearest hospital [64]. Therefore, it is important to know the total population that is able to access the nearest hospital for cardiac services within one hour.

Based on our understanding, we try to find answers the following questions-

- What are patient counts for important categories? (provision)
- What is time taken by people in the district to reach hospital for cardiac services? (reach)
- What is the total counts of patients availing treatments in hospitals in different time-periods for Khammam district? (provision + access + reach)
- What is the total counts of patients availing treatments in different time-periods for all mandals? (provision + access + reach)
- What is the number of patients availing gynecology and obstetrics services in Area Hospital-Sathupally? (provision + access + reach)
- What is the total number of patients availing Polytrauma and Nephrology services in different mandals in the years 2013 and 2014? (provision + access + reach)

Multiple visualizations can be generated by the platform using combinations of different attributes, time-periods and geo-locations. Attributes displayed on charts could be single, multiple or aggregated. Similarly, time-period shown could be single, multiple or aggregated. The geo-location could be a single point or multiple points. In this case study, we show the results obtained on generating visualizations for some of these combinations of attribute, time-period and geo-location.

6.5 Case Study Results

In order to demonstrate that visualizations can reveal patterns and trends easily compared to only statistical analysis, tables corresponding to each of them have been shown along with the questions can be answered easily through the visualization generated by the platform.

6.5.1 Aggregated Attributes, Aggregated over Time-period and Single Geo-location

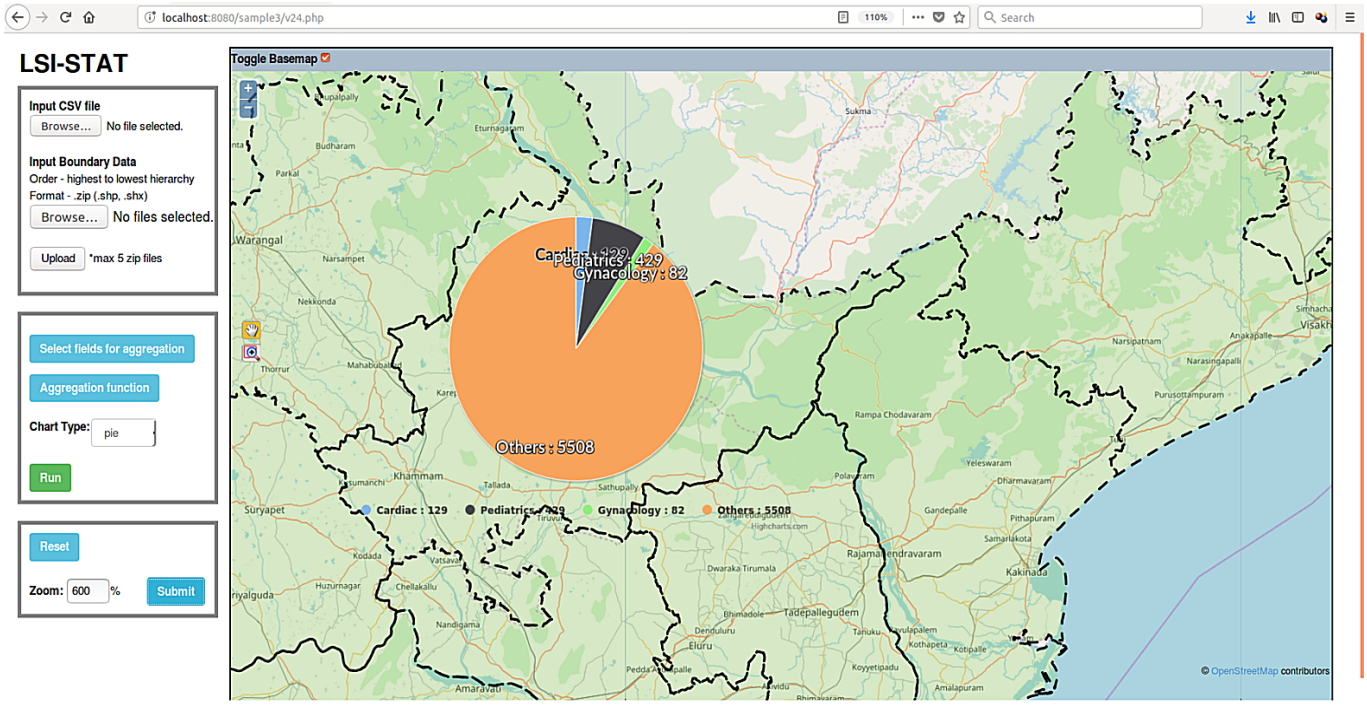

Figure 6.7: Counts of patients availing treatment in different hospitals for all categories in Khammam district

Figure 6.7 shows the counts of patients availing treatment in different hospitals for all categories in Khammam district. The visualization helps us answer the following questions.

Among categories- Pediatrics, Cardiac and Gynecology which had the highest number of patients?
Pediatrics (429)

What is the order of number of patients from maximum to minimum?
Pediatrics (429), Cardiac (129) and Gynecology (82)

What is the percentage share of patients from these three categories together?
Less than 25%

Category	Patient Counts
Cardiac	129
Pediatrics	429
Gynecology	82
Others	5508

Table 6.1: Counts of patients availing treatment in different hospitals for all categories in Khammam district

6.5.2 Multiple Attributes, Aggregated over Time-periods and Single Geo-location

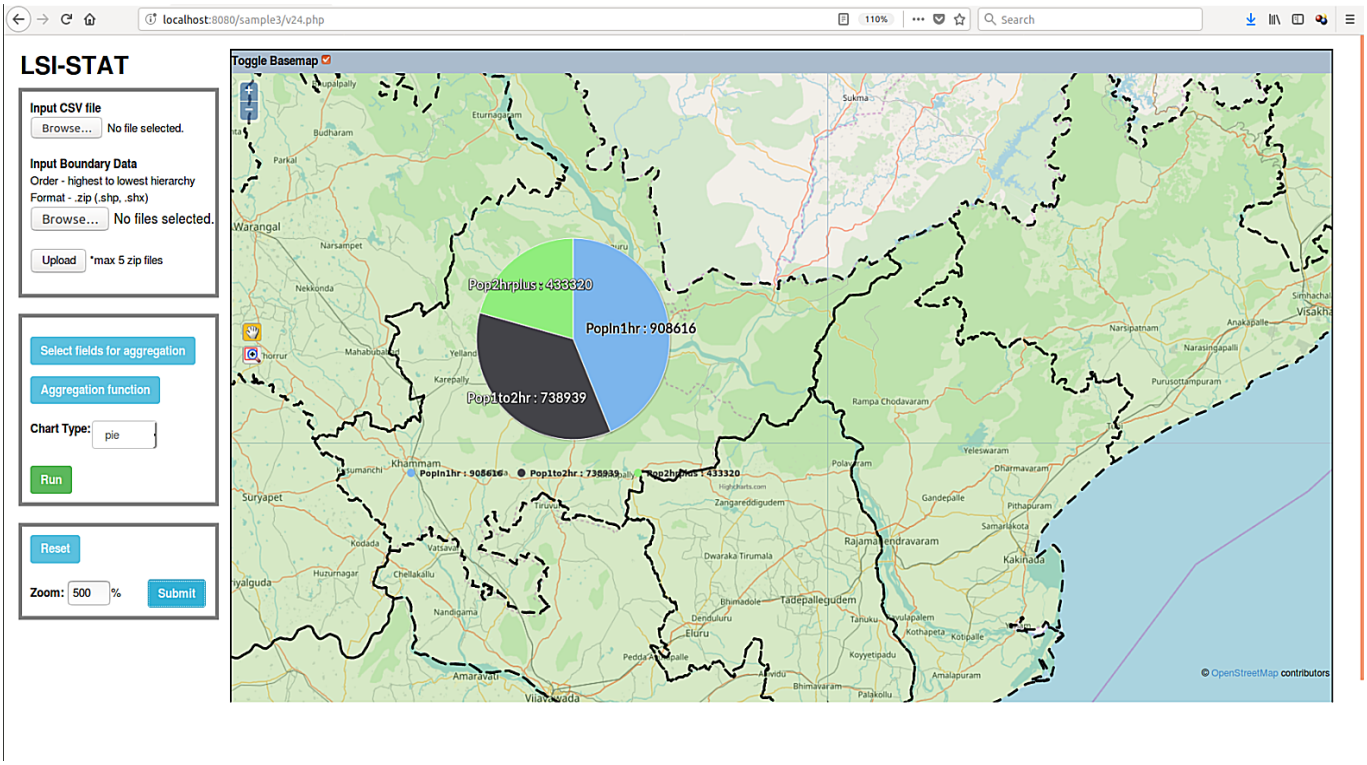

Figure 6.8: Time taken to reach Mamatha hospital for cardiac services in Khammam district

Figure 6.8 shows the time taken to reach Mamatha hospital for cardiac services in Khammam district. The visualization helps us answer the following questions.

What is the time taken to travel for cardiac services for most of the population in Khammam district?

In less than 1 hour

What is the percentage of the population which can travel in less than 1 hour?

Between 25% to 50%

What are the time-periods taken by the population to reach hospital for cardiac services?

In less than one hour (908616), 1 to 2 hours (738939), more than 2 hours (433320)

Time taken for population to reach	Population Count
Less than 1 hour	908616
In 1 to 2 hours	738939
In more than 2 hours	433320

Table 6.2: Time taken to reach Mamatha hospital for cardiac services in Khammam district

6.5.3 Multiple Attributes, Multiple Time-periods and Single Geo-location

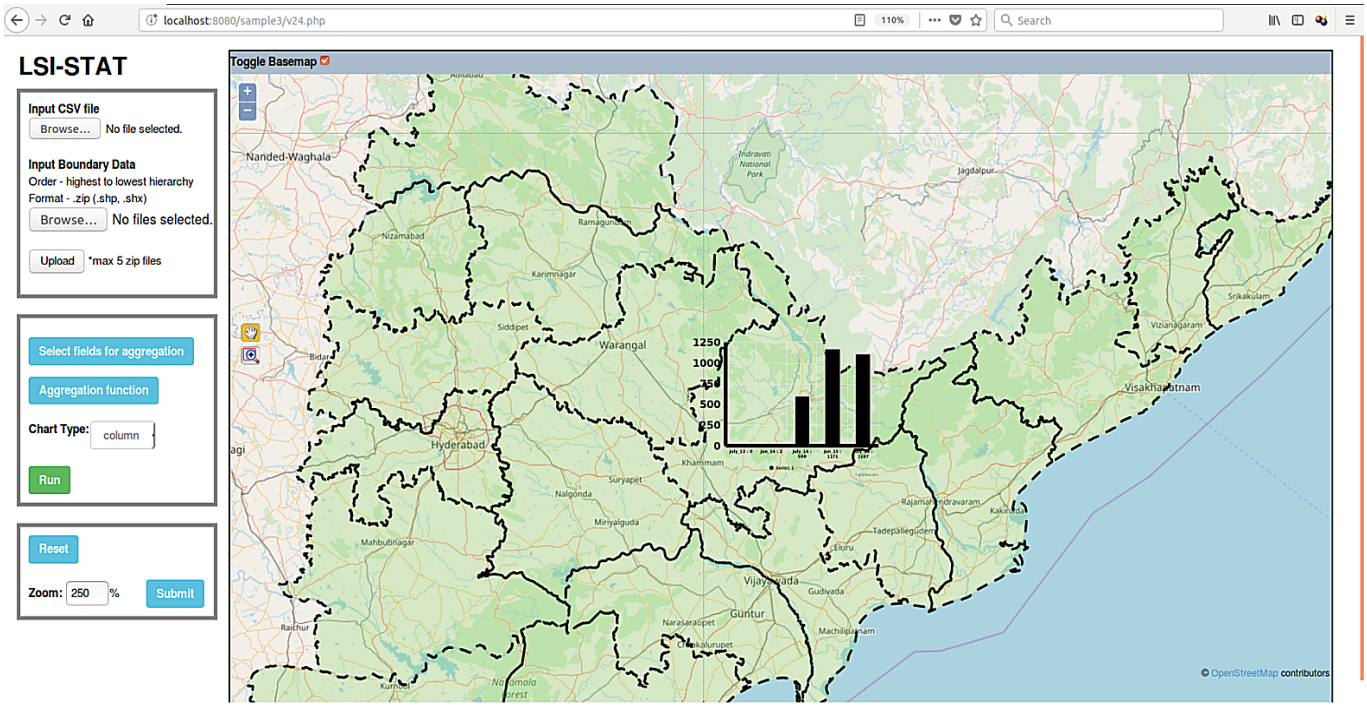

Figure 6.9: Total patients availing treatment in different time-periods for Khammam district

Figure 6.9 shows the total patients availing treatment in different time-periods for Khammam district. The visualization helps us answer the following questions.

What is the trend observed?

The number of people going to hospitals for treatment has increased over the time and then slightly decreased towards the end.

During which time-period did most people avail treatment?

July 2015-Jan 2016

Time-period	Population Count
July 2013 - Jan 2014	0
Jan 2014 - July 2014	2
July 2014 - Jan 2015	598
Jan 2015 - July 2015	1171
July 2015 - Jan 2016	1107

Table 6.3: Total patients availing treatment in different time-periods for Khammam district

6.5.4 Multiple Attributes, Multiple Time-periods and Multiple Geo-locations

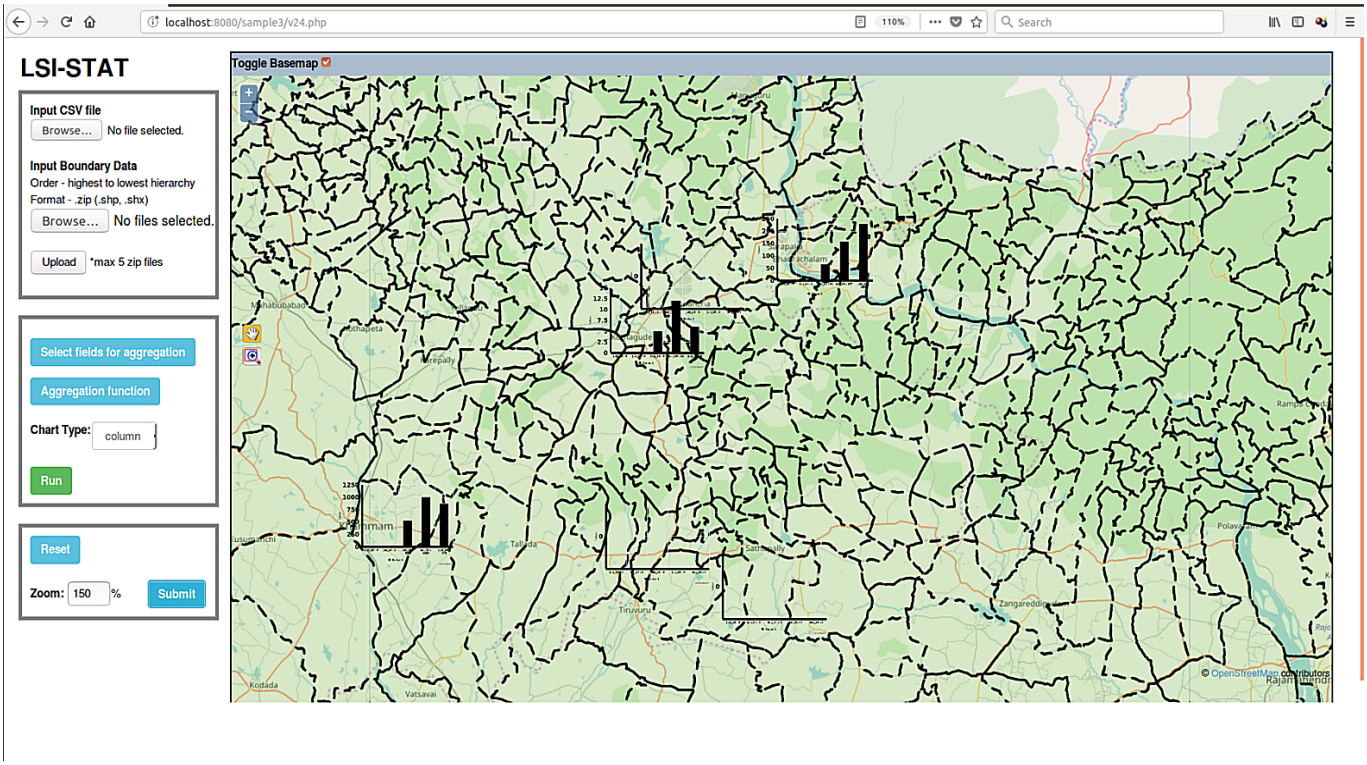

Figure 6.10: Total patients availing treatment in different time-periods for mandals in Khammam district

Figure 6.10 shows the total patients availing treatment in different time-periods for mandals in Khammam district. The visualization helps us answer the following questions.

Are patients for all mandals availing treatment in these hospitals?

No

Is the trend same across all mandals?

No, not the same

Is the trend identical similar in some mandals?

The trend is same for two of the mandals. The number of people going to these hospitals for treatment has increased over the time and then slightly decreased towards the end.

Are these two mandals neighbors?

These two mandals are in the same neighborhood but are not neighbors.

What is the other trend observed?

The number of people going to these hospitals for treatment has continuously increased over time.

July'13	Jan'14	July'14	Jan'15	July'15
0	0	5	12	6
0	1	527	1002	872
0	1	66	157	229

Table 6.4: Total patients availing treatment in different time-periods for mandals in Khammam district

6.5.5 Aggregated Attribute, Single Time-period and Multiple Geo-locations

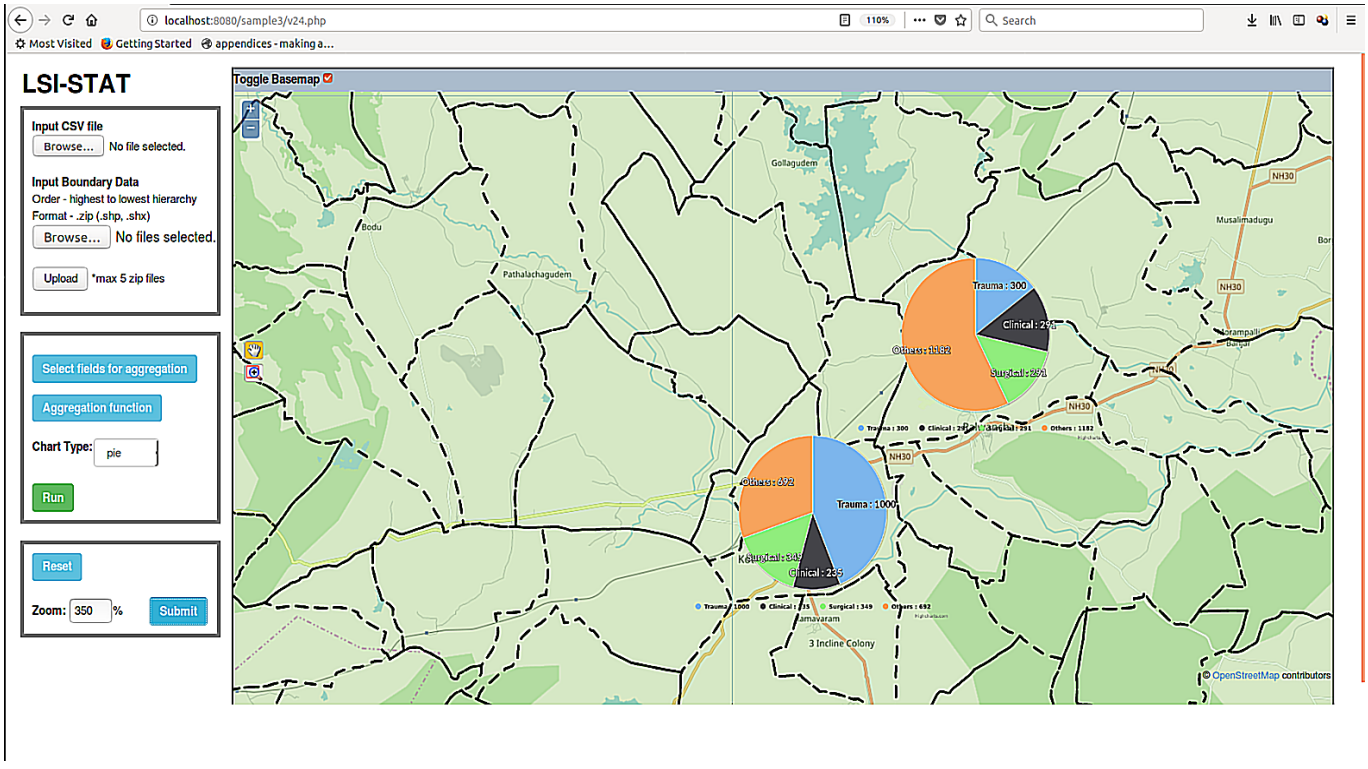

Figure 6.11: Patients availing Trauma, Surgical, Clinical and Other services in different mandals

Figure 6.11 shows the patients availing Trauma, Surgical, Clinical and Other services in different mandals. The visualization helps us answer the following questions.

Which type of services are availed by maximum number of patients in hospital in Kothagudem?
Maximum number of people avail Trauma services in hospital in Kothagudem (1000).
What is the order of services availed by patients in Kothagudem (from highest to lowest number of patients)
The order is Trauma, followed by Others, Surgical and Clinical services
Is the order similar in neighboring mandal?
In neighboring mandal - Paloncha, the order is different. The order is Others, Trauma, Clinical and Surgical (from highest to lowest number of patients) services. The number of patients for Clinical and Surgical services are equal.

Hospital	Trauma	Clinical	Surgical	Others
Bhadrachalam	2911	1145	635	319
Kothagudem	1000	235	349	692
Penuballi	9112	450	354	193
Sathupally	530	719	1129	912
Paloncha	300	291	291	1182
District	1111	1145	222	212
Jaya	29	114	935	919
Mamatha	113	114	356	192
Raksha	211	615	935	1675
Sriram	1411	1445	1235	1679
Srujan	911	945	435	419

Table 6.5: Patients availing Trauma, Surgical, Clinical and Other services in hospitals in different mandals

Chapter 7

Conclusion

In this thesis, we are addressing the question of how to build a web-based spatio-temporal interactive analytical platform which generates dynamic visualizations and captures both spatial and temporal variations. We capture these variations for each datapoint with the help of charts.These charts are generated during processing and stored in chart cache. Spatial aggregation is required to be computed each time during processing. This requires a large number of operations to be performed and increases the time taken for data visualization. In order to minimize the time taken, we compute spatial aggregation using the user-uploaded data, boundary files and develop spatial models during pre-processing. Later, during processing, these spatial models are used along with user selected fields to generate charts which are then stored as per the spatial-level or hierarchy. The main idea behind storing it in chart cache is that for any change in geo-hierarchy order of boundary files or modification of one of the boundary files, there is no need to recompute models and re-generate charts for the rest of the boundary files.

LSI-STAT uses geometry based aggregation which is different from other pre-existing tools to drill-up and drill-down. In addition to this, the levels in the hierarchy are not required to be present in the dataset. A completely different geohierarchy can be defined by the user from the geographical levels present in the input csv file. The tool is more useful for GIS Users since shapefiles are frequently used by them.

We try to capture some of the important and required features such as supporting multiple chart types on map, visualization with respect to multiple variables, spatial aggregation, user-defined geo-hierarchy, zoom by area, displaying charts in combination with maps etc. Unnecessary features complicate the user interface. We attempt to develop a unique preprocessing based tool with chart cache which captures spatial aggregation along with its neighborhood. Spatial aggregation is computed by the tool based on geometry.

Spatial aggregation is computed during preprocessing. Moreover, spatial model is developed such that it is not required to be recomputed each time any shapefile is updated or geo-hierarchy order is changed. In our approach, we use chart cache to store charts so that they dont need to be regenerated each time. The preprocessing component used is geojson file and phantomjs server, spatial indexing file

and chart cache are the processing components used by this platform. In this way, LSI-STAT is different from other existing tools.

The tool provides various interactive controls to the user such choosing chart type, aggregation type, fields for performing aggregation, panning, zooming, zoom by area, zoom charts and reset.

However, this tool has its own limitations. It requires latitude and longitude to be given by the user. Data input is restricted to csv and shapefiles. The tool works well for smaller volumes of data but for larger volumes, the processing and interaction time increases drastically. The user can upload only upto five geo-levels of shapefiles at a time.

The developed tool can be enhanced by adding other features such as time slider, word cloud, supporting multiple data input formats and reducing boundary file rendering time by using tile-based approach instead of geojson.

Two cases studies done using this tool successfully demonstrate its application and utility to the end-users. It can be seen that the visualizations generated by the platform are very intuitive. This makes it easy for the user to interpret and infer from visualizations. Also, many of these cannot be obtained through only statistical analysis. This can be seen from the table and images present in their case study results. Additionally, with the help of single visualization, we are able to answer many questions. Moreover, a large number of visualizations can be generated by the user using location, time and attribute combinations. Attribute could be single or multiple fields. Similarly, time could be single point, multiple time-periods or aggregated over time. Location could be a single region or multiple regions.

Bibliography

[1] Keith C. Clarke. Geog183: Cartographic Design and Geovisualization, Spring Quarter, Lecture 3. http://www.geog.ucsb.edu/ kclarke/Geography183/Lecture03.pdf. 2018 [Accessed: 15- Feb- 2018]

[2] Friendly, M., Denis, D.J.*Milestones in the history of thematic cartography, statistical graphics, and data visualization*. 2001.

[3] Alan M. MacEachren and Menno-Jan Kraak. Research challenges in geovisualization. *Cartography and Geographic Information Science*, 28(1), 2001.

[4] Chaudhuri, S., Dayal, U. An overview of data warehousing and OLAP technology. *ACM SIGMOD* Rec. 26(1), 6574 (1997).

[5] Bdard, Y., Gosselin, P., Rivest, S., Proulx, M. J., Nadeau, M., Lebel, G., and Gagnon, M. F. (2003). Integrating GIS components with knowledge discovery technology for environmental health decision support. *International Journal of Medical Informatics*, Vol. 70, No. 1, pp. 7994.

[6] Tutorialspoint. Data Warehousing OLAP. https://www.tutorialspoint.com/dwh/dwh_olap.htm. [Accessed: 15- Feb- 2018]

[7] Sonia, R., Yvan, B., and Pierre, M. (2001). Toward better support for spatial decision making: Defining the characteristics of Spatial Online Analytical Processing (SOLAP). *Geomatica*, Vol. 55, No. 4, pp. 539-555.

[8] Rivest, S., Bdard, Y., Proulx, M.J., Nadeau, M., Hubert, F., Pastor, J. SOLAP technology: merging business intelligence with geospatial technology for interactive spatio-temporal exploration and analysis of data. *ISPRS J. Photogramm. Remote Sens*. 60(1), 1733 (2005).

[9] Silva, J., Oliveira, A.G., Fidalgo, R.N., Salgado, A.C., Times, V.C. *Modelling and querying geographical data warehouses. Inf. Syst.* 35(5), 592614 (2010).

[10] Wikipedia. Spatial Relation. https://en.wikipedia.org/wiki/Spatial_relation. [Accessed: 15- Feb- 2018]

[11] Openlayers. https://openlayers.org. [Accessed: 8- Jun- 2017]

[12] Alam M. Maceachren and Dr. Fraser Taylor, Modern Cartography Series, Academic Press, 1994, Volume 2, Pages 1-12 [13] Choropleth map. https://en.wikipedia.org/wiki/Choropleth_map [Accessed:

15- Feb- 2018]

[14] Create choropleth map using openlayers? https://gis.stackexchange.com/questions/276918/create-choropleth-map-using-openlayers. [Accessed: 15- Feb- 2018]

[15] Wikimedia Commons. https://commons.wikimedia.org/wiki/File:WOA09_sea-surf_SAL_AYool.png [Accessed: 15- Feb- 2018]

[16] The Data Visualization Catalog. Treemap. https://datavizcatalogue.com/methods/treemap.html. [Accessed: 15- Feb- 2018]

[17] The Data Visualization Catalog. Dotmap. https://datavizcatalogue.com/methods/dot_map.html. [Accessed: 15- Feb- 2018]

[18] HERE. Heatmap. https://developer.here.com/documentation/geovisualization/topics/concept-heatmaps.html [Accessed: 15- Feb- 2018]

[19] DeBoer M. Understanding the heat map. *Cartographic Perspectives*, No. 80, pp. 3943. 2015

[20] Socialcops. 7 Techniques to visualize geospatial data. https://blog.socialcops.com/academy/resources/7-techniques-to-visualize-geospatial-data. [Accessed: 15- Feb- 2018]

[21] NoSQL. https://www.mongodb.com/nosql-explained. [Accessed: 15- Feb- 2018]

[22] Steve Andriole. Unstructured Data: The Other Side of Analytics. Forbes. http://www.forbes.com/sites/steveandriole/2015/03/05/the-other- side-of-analytics. [Accessed: 15- Feb- 2018]

[23] M. Tanwar, R. Duggal, S.K. Khatri. Unravelling unstructured data: A wealth of information in big data, in: 2015 *4th International Conference on Reliability, Infocom Technologies and Optimization, ICRITO, Trends and Future Directions*, 2015, pp. 16.

[24] GeoVISTA Center. GeoVisual Analytics. https://www.geovista.psu.edu/research/projects/geovisualanalytics.html. [Accessed: 15- Feb- 2018]

[25] Cartography and Geovisualization Group. http://www.cartography.oregonstate.edu. [Accessed: 15- Feb- 2018]

[26] Center for Geospatial Analytics. https://cnr.ncsu.edu/geospatial/about. [Accessed: 15- Feb- 2018]

[27] Mehdi Dastani, The Role of Visual Perception in Data Visualization, *Journal of Visual Languages & Computing*, Volume 13, Issue 6, 2002, Pages 601-622.

[28] N. Tryfona and M. Egenhofer. Consistency among parts and aggregates: a computational model. *Transactions in GIS*, 4(3):189206, 1997.

[29] Garg, Nipun and Surabhi Mithal. *CSCI 8715 Spatial databases*. (2011).

[30] Malinowski, E., & Zimnyi, E. (2005, July). Spatial hierarchies and topological relationships in the spatial MultiDimER model. *In British National Conference on Databases* (pp. 17-28). Springer, Berlin, Heidelberg.

[31] GeoKettle. http://www.spatialytics.org/projects/geokettle. [Accessed: 15- Feb- 2018]

[32] Spatialytics Projects. http://www.spatialytics.org/projects. [Accessed: 15- Feb- 2018]

[33] GeoMondrian. http://www.spatialytics.org/projects/geomondrian. [Accessed: 15- Feb- 2018]

[34] Tableau Software https://en.wikipedia.org/wiki/Tableau_Software. [Accessed: 27- Feb- 2018]

[35] Tableau Training https://www.tableau.com/learn/training. [Accessed: 27-Feb- 2018]

[36] Tableau Community http://onlinehelp.tableau.com/current/pro/desktop/en-us/help.htm#reference_overview.htm. [Accessed: 27- Feb- 2018]

[37] GeoVista CrimeViz https://www.geovista.psu.edu/CrimeViz. [Accessed: 27- Feb- 2018]

[38] LANDIS-II http://www.landis-ii.org. [Accessed: 10- Mar- 2018]

[39] LANDIS-II Extensions http://www.landis-ii.org/extensions. [Accessed: 10- Mar- 2018]

[40] Northern Research Station Tools https://www.nrs.fs.fed.us/tools/landis. [Accessed: 10- Mar- 2018]

[41] LANDIS-II Visualization Tool https://www.youtube.com/watch?v=N63Bd5eyDDw. [Accessed: 10- Mar- 2018]

[42] EWGAT Codebase https://github.com/XiancaiTian/EWGAT. [Accessed: 10- Mar- 2018]

[43] EWGAT Demo https://www.youtube.com/watch?v=-bJpOlG0z2s. [Accessed: 10- Mar- 2018]

[44] AFDC TransAtlas https://maps.nrel.gov/transatlas. [Accessed: 10- Mar- 2018]

[45] Wikipedia. OpenLayers. https://en.wikipedia.org/wiki/OpenLayers. [Accessed: 15- Mar- 2018]

[46] Wikipedia. Cite_Note3 https://en.wikipedia.org/wiki/OpenLayers#cite_note-3. [Accessed: 15- Mar- 2018]

[47] Wikipedia. Cite_Note4 https://en.wikipedia.org/wiki/OpenLayers#cite_note-4. [Accessed: 15- Mar- 2018]

[48] Wikipedia. Apache_Tomcat https://en.wikipedia.org/wiki/Apache_Tomcat. [Accessed: 15- Mar- 2018]

[49] MuleSoft. Apache Tomcat Resources. https://www.mulesoft.com/tcat/understanding-apache-tomcat [Accessed: 15- Mar- 2018]

[50] Highcharts Blog. https://www.highcharts.com/blog/products/highcharts/ [Accessed: 15- Mar- 2018]

[51] HackerNoon. 7 Ways Predictive Analytics is a Boon to Retail Industry. [Accessed: 15- Mar- 2018] https://hackernoon.com/7-ways-predictive-analytics-is-a-boon-to-retail-industry-645832730926

[52] Sales Jan 2009. https://bigml.com/user/czuriaga/gallery/dataset/555c67beaf447f4a73001457. [Accessed:28- Jun- 2017]

[53] Thematic Mapping- World Borders Dataset. http://thematicmapping.org/downloads/world_borders.php. [Accessed: 28- Jun- 2018]

[54] ArcGIS Continent Shapefile. http://www.arcgis.com/home/item.html?id=3c4741e22e2e4af2bd4050511b9fc6ad [Accessed: 28- Jun- 2018]

[55] Global Administrative Areas. http://www.gadm.org/download [Accessed: 28- Jun- 2018]

[56] SIGeGov. www.csi-sigegov.org/ppts2/Aarogysri-%20ppt-CSI-Nihilent.pptx. [Accessed: 15- Feb- 2018]

[57] Aarogyasri Health Care Trust. http://www.aarogyasri.telangana.gov.in/documents/10181/13460/Annual_Report_201112.pdf. [Accessed: 10- Oct- 2016]

[58] Aarogyasri Health Care Trust. http://www.aarogyasri.telangana.gov.in/documents/10181/13460/Planning_Coordination_Presentation.pdf. [Accessed: 10- Oct- 2016]

[59] Worldatlas. Where is Khammam, India? https://www.worldatlas.com/as/in/tg/where-is-khammam.html. [Accessed: 15- Feb- 2018]

[60] Maps Of India. Khammam District Map (Telangana). https://www.mapsofindia.com/maps/telangana/districts/khammam.htm. [Accessed: 15- Feb- 2018]

[61] Wikipedia. Khammam district. https://en.wikipedia.org/wiki/Khammam_district. [Accessed: 10- Mar- 2018]

[62] Knowledge Research Laboratory. http://kelab.tamu.edu/standard/restoration/landis/LANDIS-II%20Model%20Description.pdf [Accessed: 10- Mar- 2018]

[63] Shalini Kanuganti, A.K. Sarkar, Ajit Pratap Singh. Quantifying Accessibility to Health Care Using Two-step Floating Catchment Area Method (2SFCA): A Case Study in Rajasthan, *Transportation Research Procedia*, Volume 17, 2016, Pages 391-399.

[64] Brual, Janette et al. Drive Time to Cardiac Rehabilitation: At What Point Does It Affect Utilization? *International Journal of Health Geographics (2010).* 27. PMC. Web. 14June 2017.

[65] Google Maps (2018). https://www.google.co.in/maps. [Accessed: 8- Jun- 2017]

[66] Apache Tomcat (2017). http://tomcat.apache.org. [Accessed: 8- Jun- 2017]

[67] Highcharts (2017). http://www.highcharts.com. [Accessed: 8- Jun- 2017]

[68] JQuery (2017). http://jquery.com. [Accessed: 8- Jun- 2017]

[69] PHP Tutorial. https://www.w3schools.com/php. [Accessed: 8- Jun- 2017]

[70] Jianghui Ying, Denis Gracanin, and Chang-Tien Lu. Web Visualization of Geo-Spatial Data using SVG and VRML/X3D. *Proceedings of the Third International Conference on Image and Graphics (ICIG04)*.

[71] Natalia Andrienko, Gennady Andrienko, and Peter Gatalsky. Exploratory spatio-temporal visualization: an analytical review. *Journal of Visual Languages & Computing*, 14(6):503 541, 2003.

[72] Diansheng Guo, Jin Chen, Alan M. MacEachren, and KeLiao. A visualization system for space-time and multivariate patterns (vis-stamp). *IEEE Transactions on Visualization and Computer Graphics*, 12(6):14611474, November 2006.

[73] Andrienko, G., Andrienko, N., 1999. Interactive maps for visual data exploration. *International Journal of Geographical Information Science*, 13(4), pp. 355-374.

[74] Andrienko, N., Andrienko, G., Gatalsky, P., 2000. Towards Exploratory Visualization of Spatio-Temporal Data. *In: 3 rd AGILE Conference on Geographic Information Science-Helsinki/Espoo*, Finland, May 25 th -27 th, 2000.

[75] Q. Ho, Q., P. Lundblad, P., T. strm, T., and M. Jern, M., 2011. A Web-Enabled Visualization Toolkit for Geovisual Analytics Visualisation and Data Analysis. *In: Proceedings of SPIE: Electronic Imaging Science and Technology*, Visualization and Data Analysis, San Francisco, USA Jan 2011.

[76] Gennady Andrienko, Natalia Andrienko, and Stefan Wrobel. Visual analytics tools for analysis of movement data. *SIGKDD Explore*. Newsletter, vol. 9, no. 2, pp. 38-46, December 2007.

[77] D. A. Keim. Information visualization and visual data mining. *Visualization and Computer Graphics, IEEE Transactions* on, vol. 8, pp. 1-8, 2002.

[78] Mao, Bo, Zhiang Wu, and Jie Cao. A framework for online spatio-temporal data visualization based on html5. *ISPRS-International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences* 39 (2012): B2.

[79] Daniel A. Keim, Christian Panse, and Mike Sips. Information visualization: Scope, techniques and opportunities for geovisualization. *Exploring Visualization*, 2005.

[80] Census of India. District census handbook for Khammam, Telangana. http://www.censusindia.gov.in/2011census. (2011). [Accessed: 10- Oct- 2016]

[81] GeoVista CrimeViz Update https://www.youtube.com/watch?v=Wj-aBS1hQ7g. [Accessed: 27- Feb- 2018]

Related Publications

- Neha Pande, KS Rajan.FOSS based Interactive Spatial Temporal Analytical Tool: a GeoBI case study of retail data Proceedings of the 38th Asian Conference on Remote Sensing:, New Delhi, India, 2017
- Neha Pande, KS Rajan. Interactive Spatio-Temporal Analytical Tool: A case study on Rajiv Aarogyasri health insurance data in Khammam district, Telangana, FOSS4G Second National Conference: ISPRS Geospatial Public Health Symposium, Hyderabad, India, July 2, 2017

Appendix A

Flowcharts

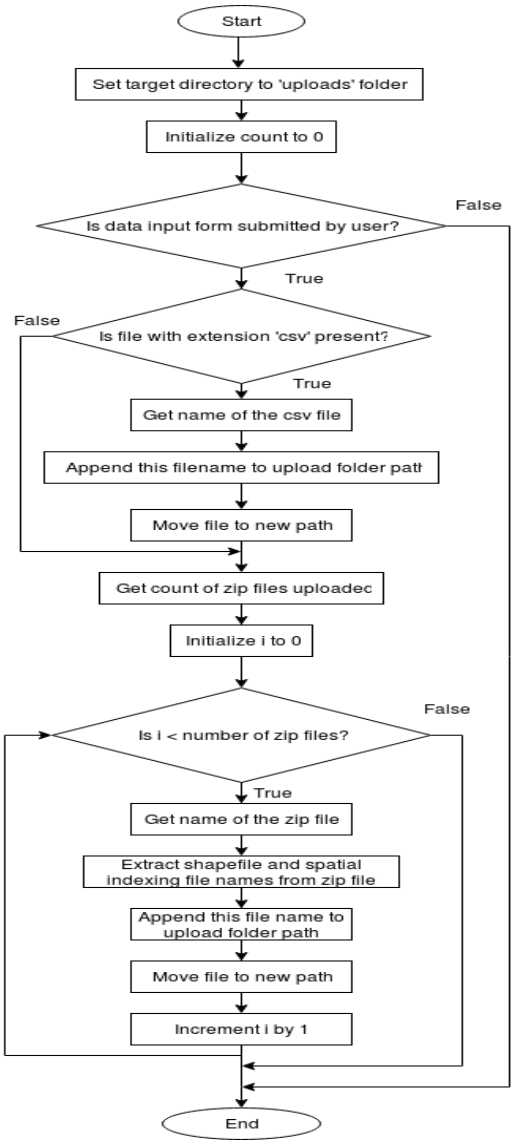

Figure A.1: Flowchart represents steps for accepting, validating and storing user input data

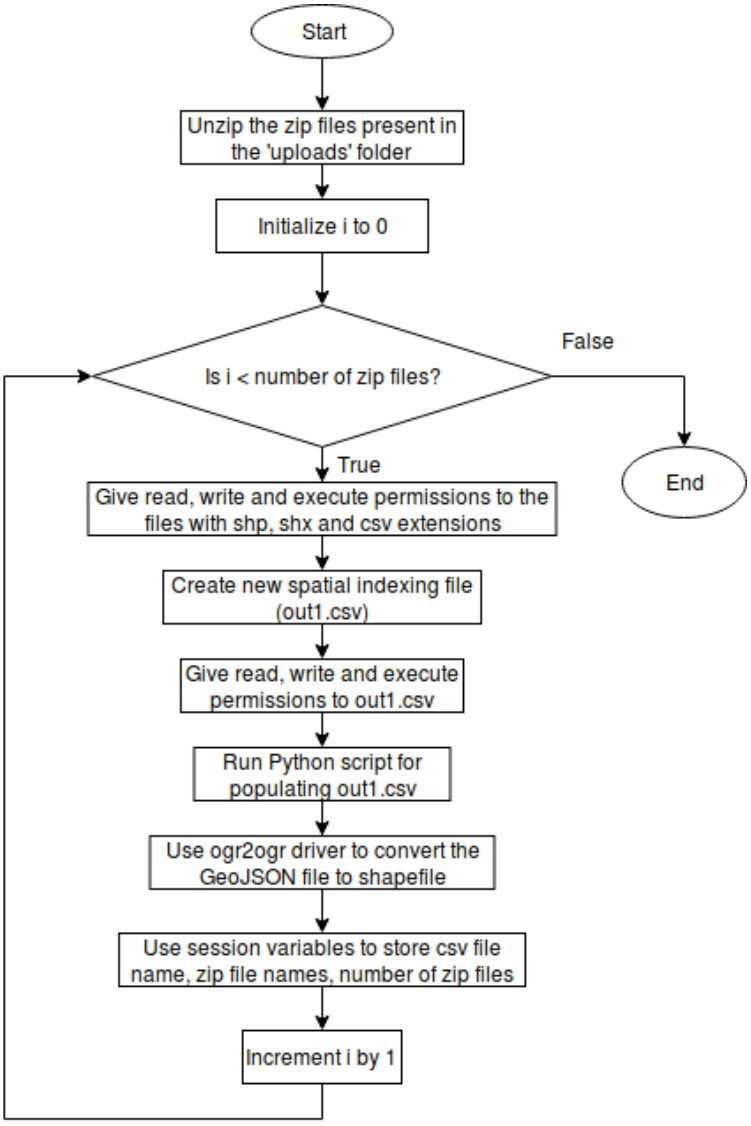

Figure A.2: Flowchart represents steps for preprocessing user uploaded files

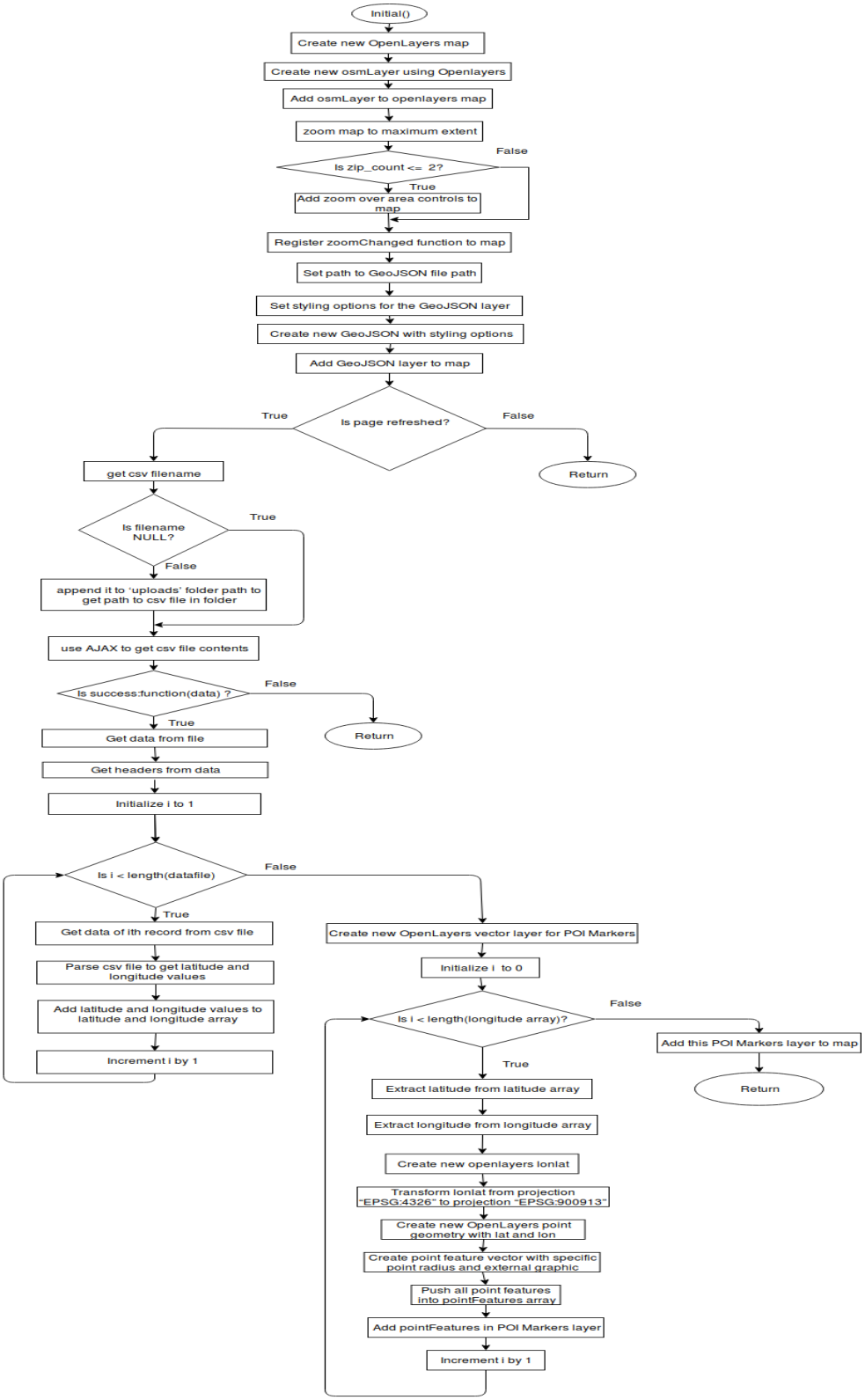

Figure A.3: Flowchart represents steps for initial setup of map panel for geovisualization

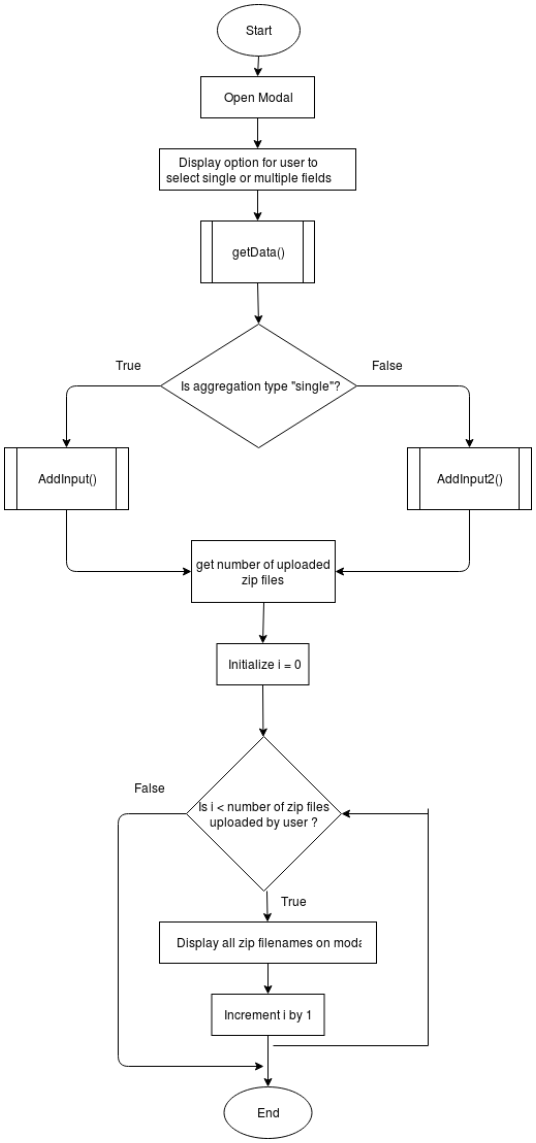

Figure A.4: Flowchart represents steps involved in invoking relevant functions based on user chosen option

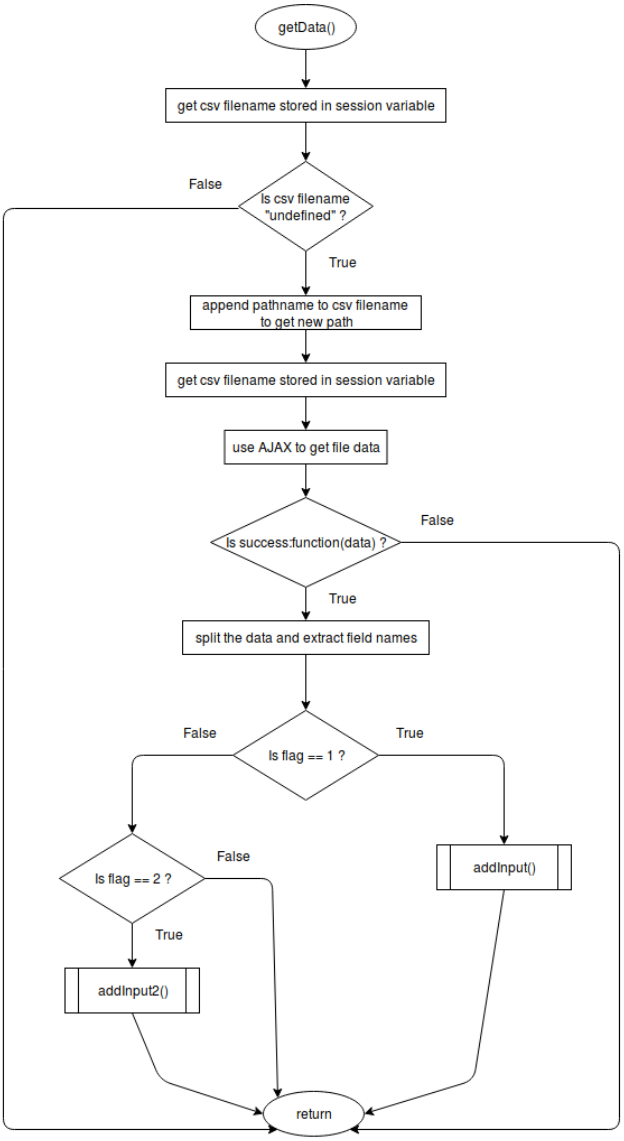

Figure A.5: Flowchart representing steps involved in extracting data from user input file for providing fields in menu

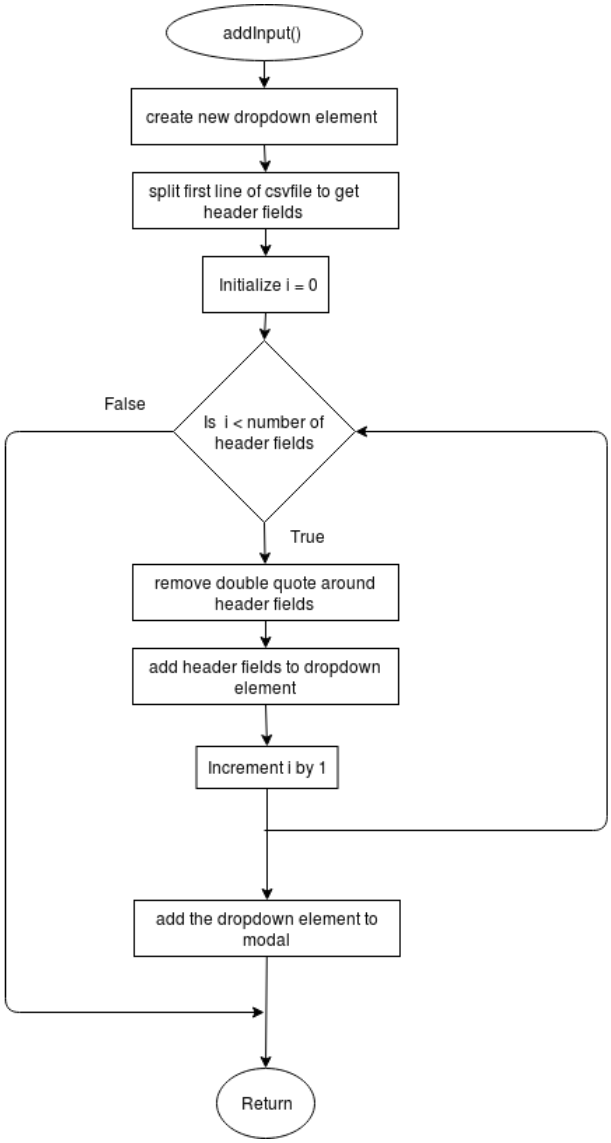

Figure A.6: Flowchart represents steps involved in providing fields to dropdown menu

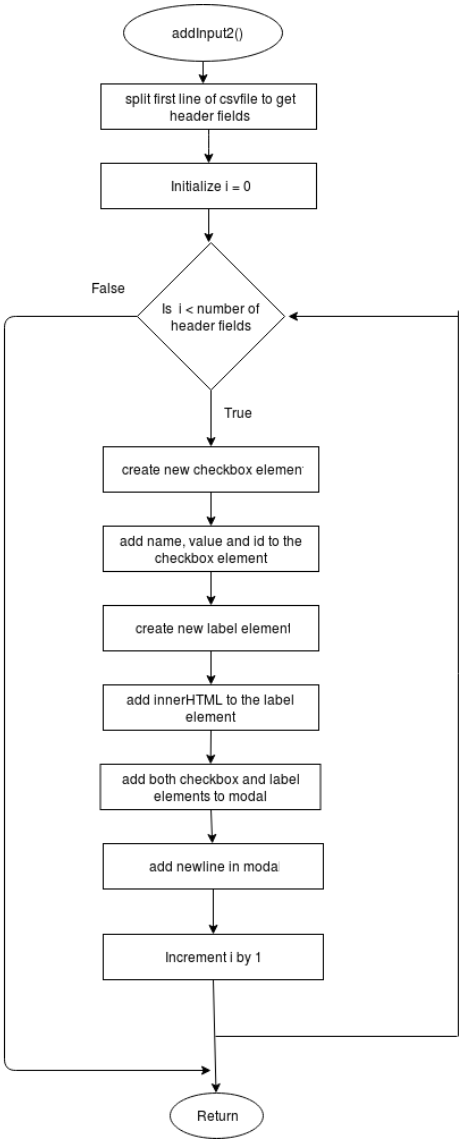

Figure A.7: Flowchart represents steps involved in providing fields as checkboxes

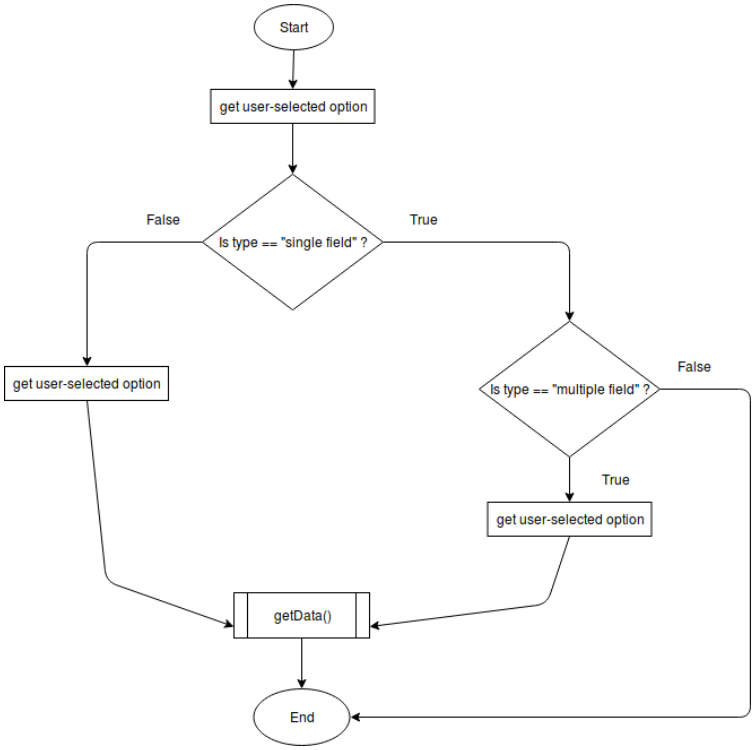

Figure A.8: Flowchart represents steps for fetching user selected option and displaying field names

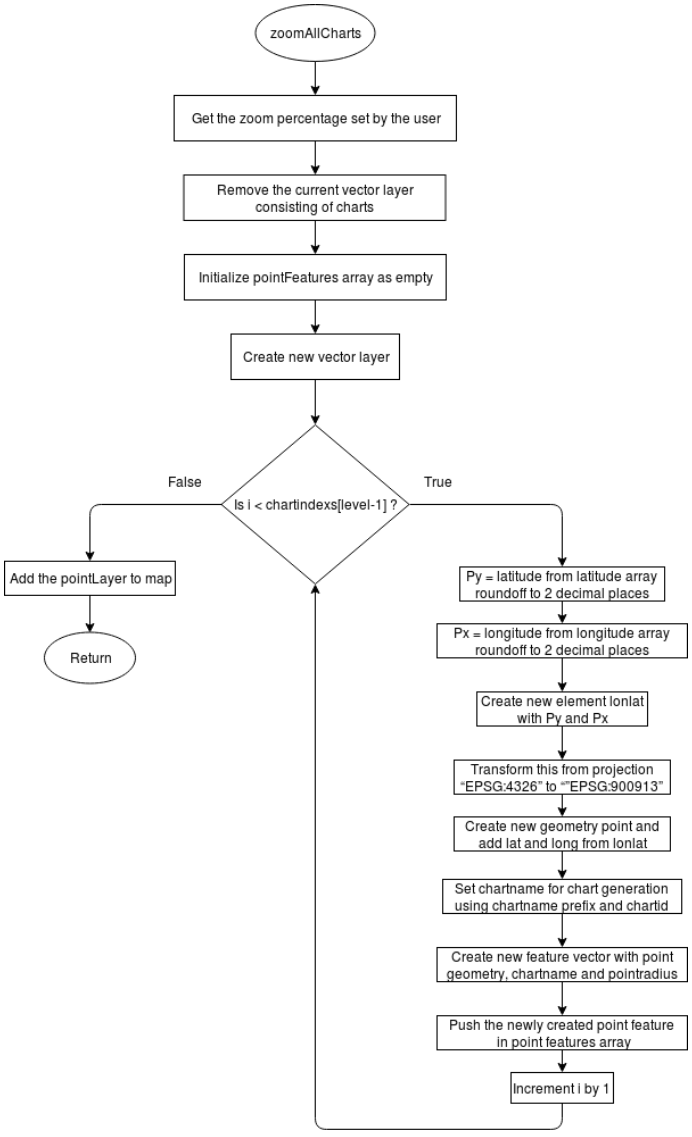

Figure A.9: Flowchart represents steps for zoom all charts by user-defined percentage

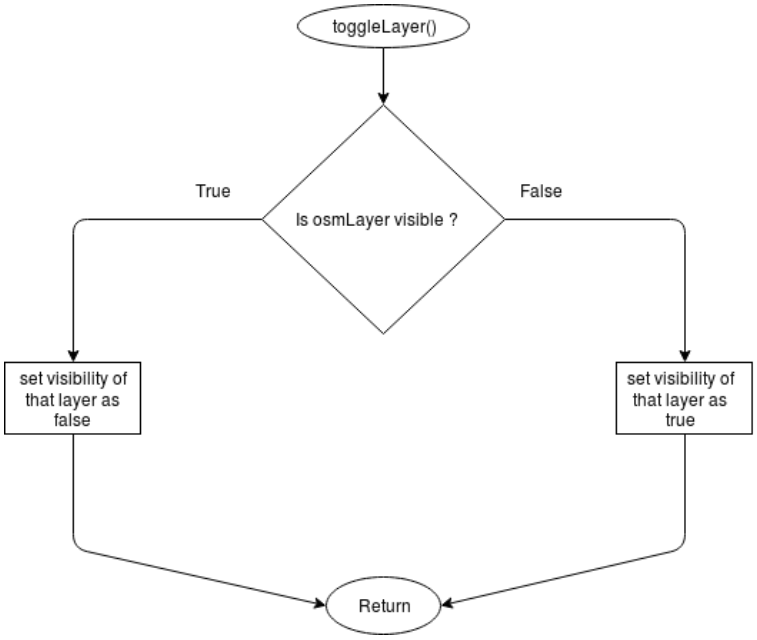

Figure A.10: Flowchart represents steps for toggling the basemap

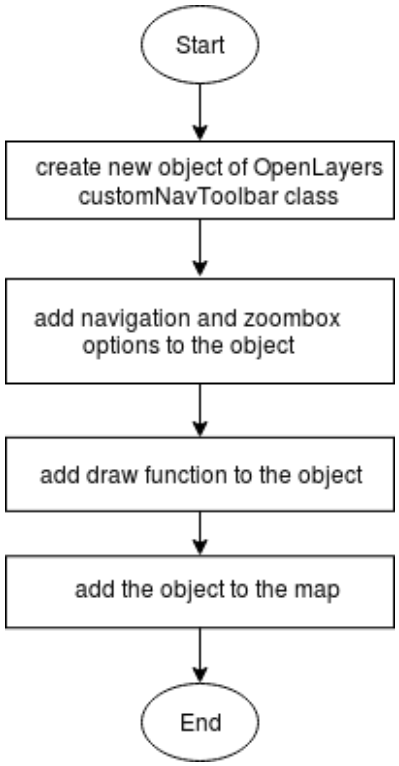

Figure A.11: Flowchart represents steps for providing Zoom by Area feature on the map

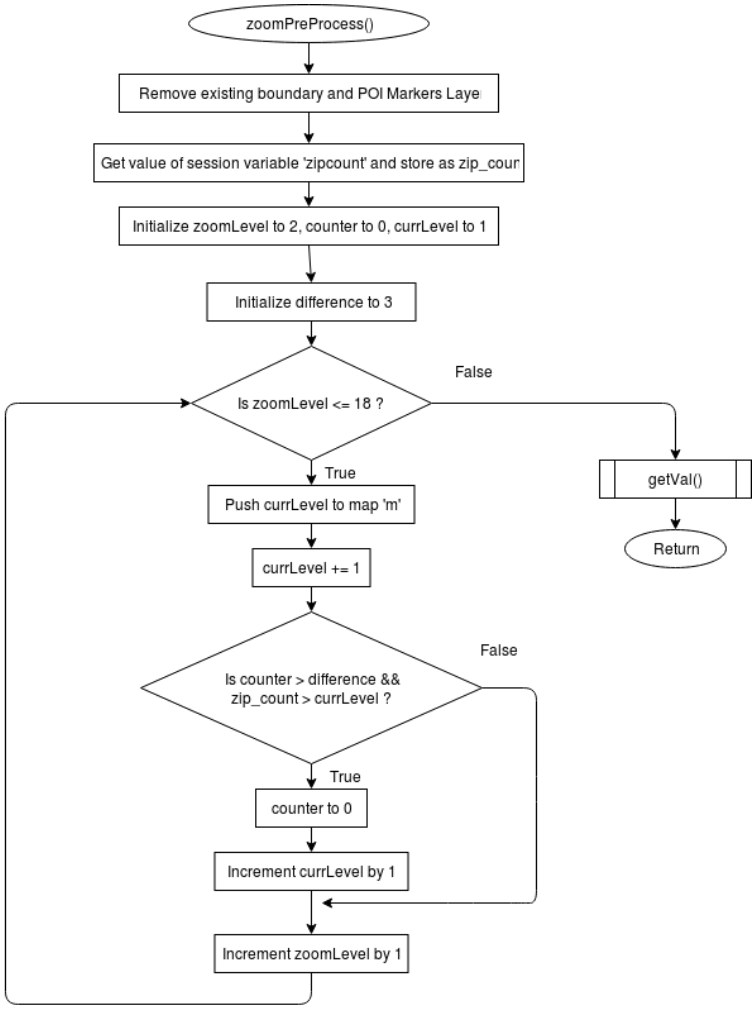

Figure A.12: Flowchart represents steps for creating a mapping between zoom level and boundary file to be displayed

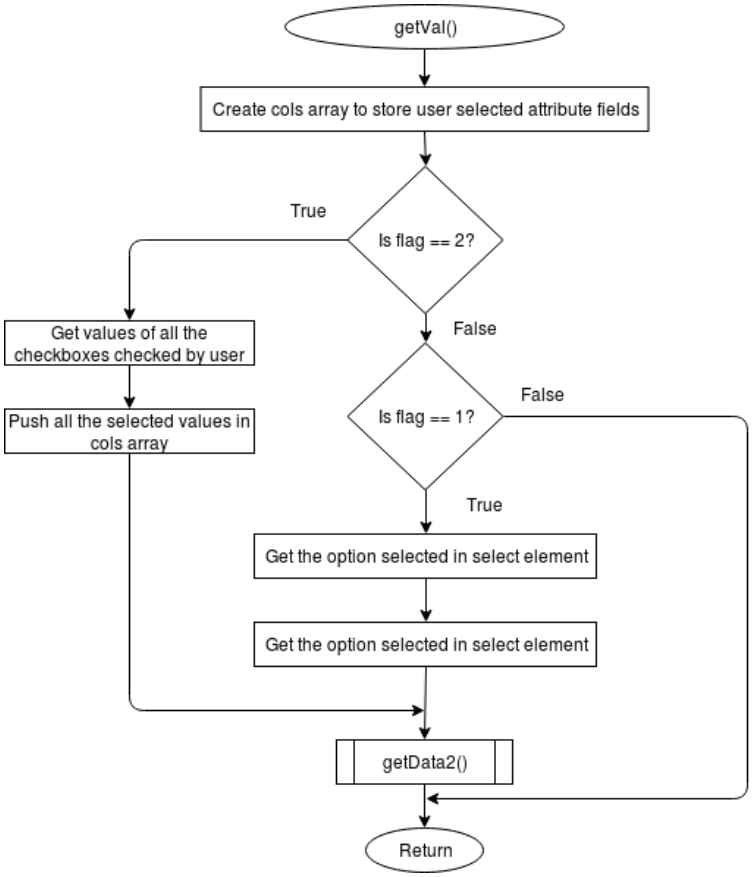

Figure A.13: Flowchart represents steps for fetching user selected fields

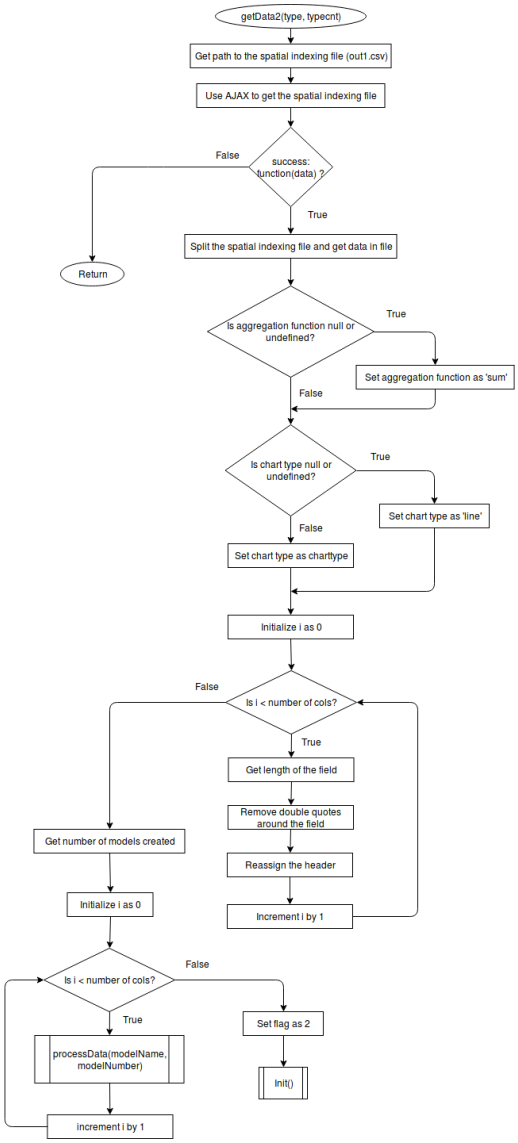

Figure A.14: Flowchart represents steps for setting chart options before chart generation

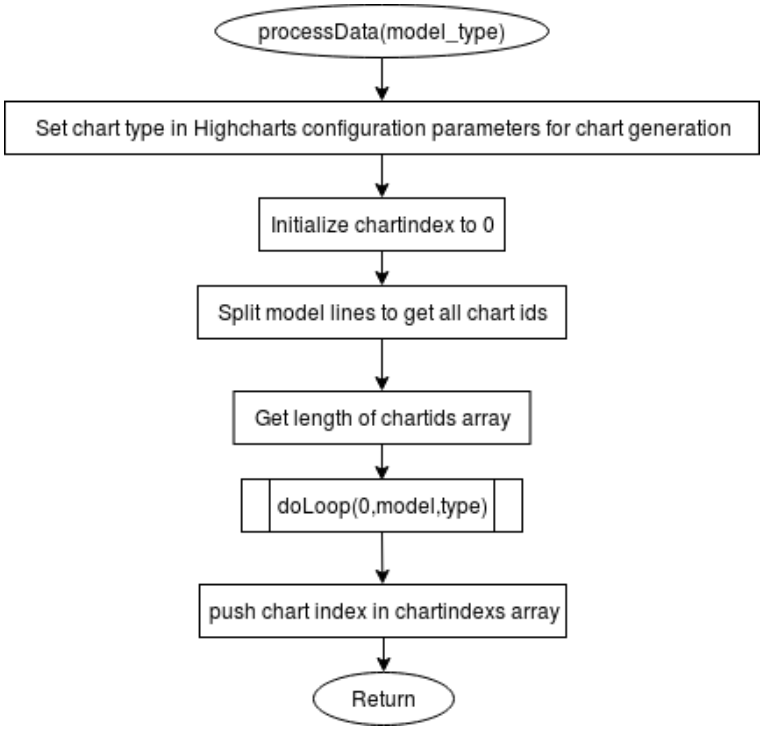

Figure A.15: Flowchart represents steps for initializing values before chart generation

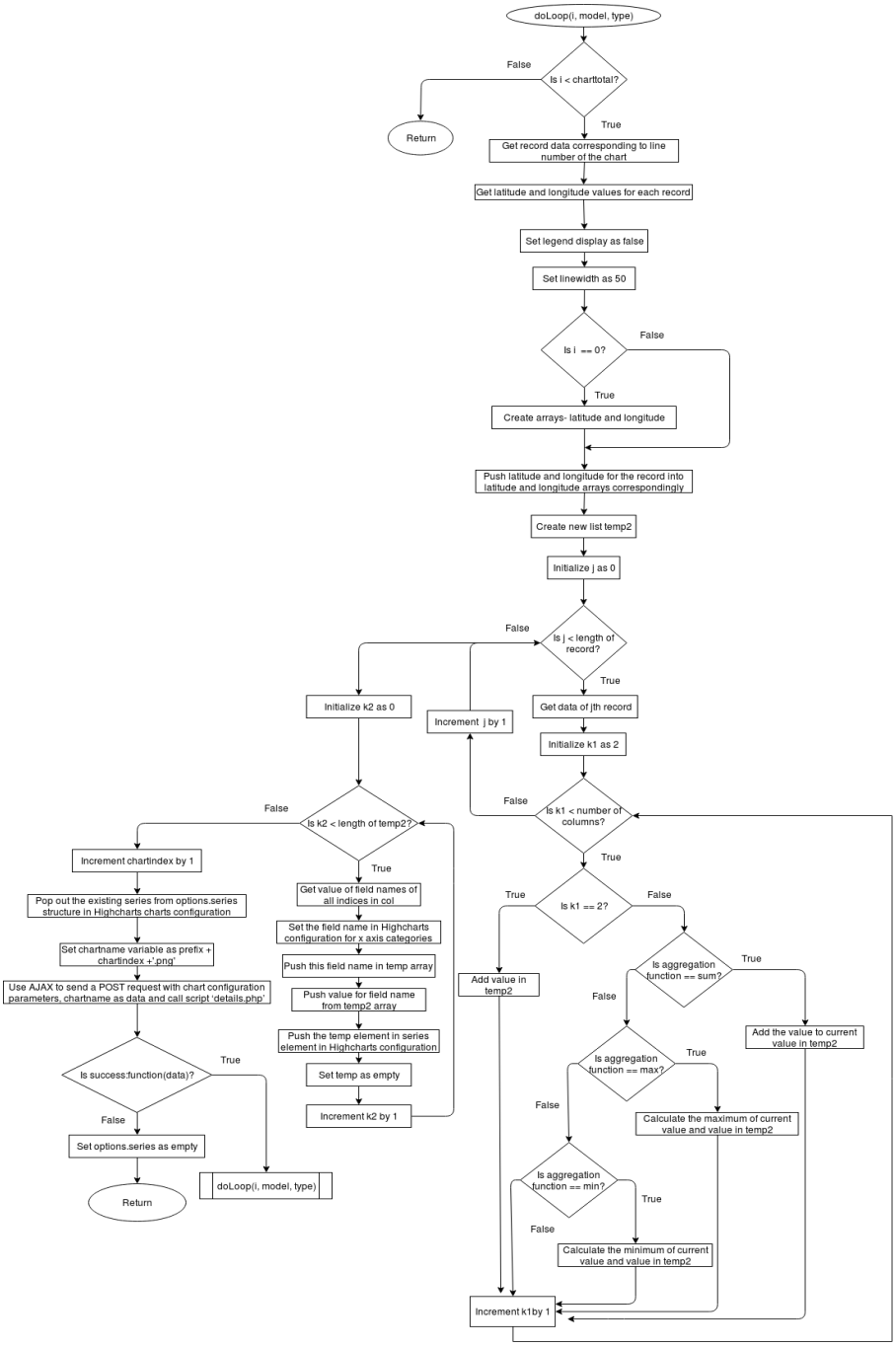

Figure A.16: Flowchart represents steps for generating charts

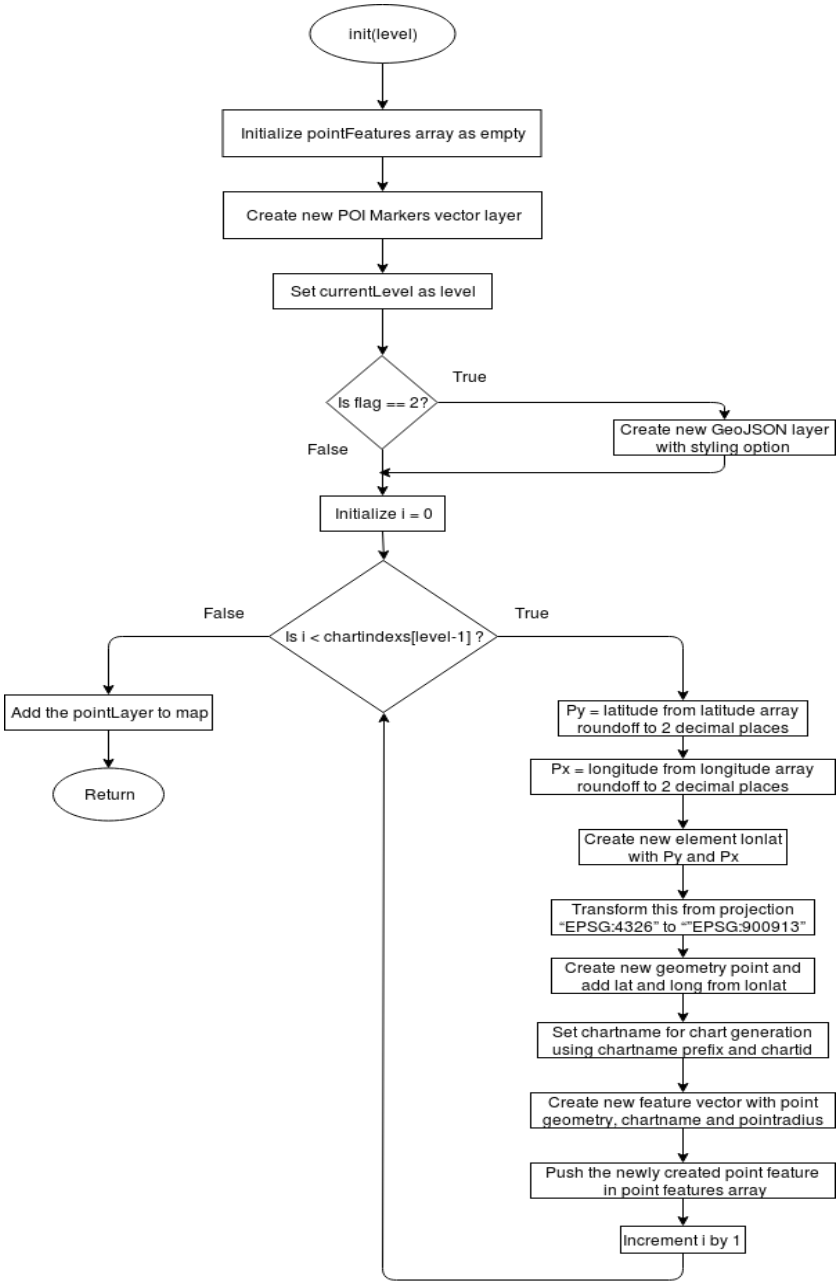

Figure A.17: Flowchart represents steps for overlaying charts on map

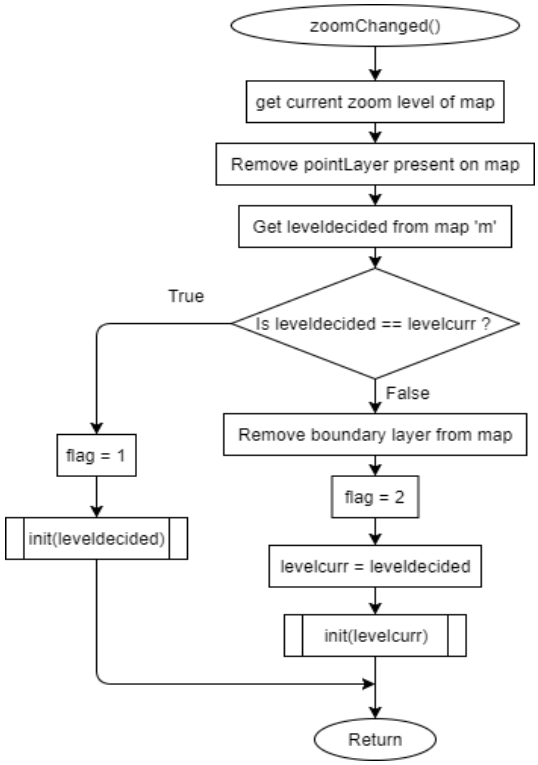

Figure A.18: Flowchart represents steps followed on change in current zoom level

Appendix B

Code Snippets

Data Upload

Listing B.1: Data Upload methods

```
<form method="post" enctype="multipart/form-data" >Input Text Data
   :</br>
  format: .csv </br>
  <input type="file" name="csv" id="csv">
    Input: highest to lowest hierarchy</br>
    format: .zip (.shp, .shx) </br>
    <input type="hidden" name="file_count" id="file_count" value
       ="0"/>
      <table id="files_table" border="0" cellpadding="0" cellspacing
         ="0">
         <tr id="new_file_row">
           <td>
              <input type="file" name="new_file[]" multiple="
                 multiple" id="new_file[0]" onchange="add_new_file(
                 this)"/>
            </td>
         </tr>
      </table>
   <input type="submit" value="Upload" name="ok" /> (max 5 zip files
      )</br>
```

```
</form>
```

```
$target_dir="./uploads/";
$cnt=0;
if(isset($_POST['ok'])) {
    if (isset($_FILES['csv'])) {
        $n = $_FILES['csv']['name'];
        $tmp = $_FILES['csv']['tmp_name'];
        $ext = pathinfo($n, PATHINFO_EXTENSION);
        $csvfile=$n;
        if (!$n) continue;
        $target_file=$target_dir.basename($n);
        move_uploaded_file($tmp, $target_file);
    }
}
```

Data Preprocessing

Listing B.2: Data Preprocessing methods

```
$number_of_files_uploaded = count($_FILES['new_file']['name']);
$uploads = $_FILES['new_file'];
for ($i = 0; $i < $number_of_files_uploaded; $i++) {
    $n = $uploads['name'][$i];
    $tmp = $uploads['tmp_name'][$i];
    $ext = pathinfo($n, PATHINFO_EXTENSION);
    $shpfile=shell_exec('cd uploads/; unzip -l '.$n.' | grep -oh "\w
       *.shp" | tr -d "[:space:]"');
    $shxfile=shell_exec('cd uploads/; unzip -l '.$n.' | grep -oh "\w
       *.shx" | tr -d "[:space:]"');
    if (!$n) continue;
    $target_file=$target_dir.basename($n);
    move_uploaded_file($tmp, $target_file);
```

```
}
```

Data Processing

Initial setup:

Listing B.3: Initial Setup of map panel

```
function initial(){
   map = new OpenLayers.Map("map");
   osmlayer = new OpenLayers.Layer.OSM();
   map.addLayer(osmlayer);
   map.zoomToMaxExtent();
   OpenLayers.Control.CustomNavToolbar = OpenLayers.Class(OpenLayers
      .Control.Panel, {
      initialize: function(options) {
         OpenLayers.Control.Panel.prototype.initialize.apply(this, [
            options]);
         this.addControls([
            new OpenLayers.Control.Navigation(),
            new OpenLayers.Control.ZoomBox({alwaysZoom:true})
         ]);
         this.displayClass = 'olControlNavToolbar'
      },
      draw: function() {
         var div = OpenLayers.Control.Panel.prototype.draw.apply(
            this, arguments);
           this.defaultControl = this.controls[0];
           return div;
         }
      });
      var panel = new OpenLayers.Control.CustomNavToolbar();
      zip_count="<?php echo $_SESSION['zipcount'] ?>";
      if(zip_count<=2)
```

```
        map.addControl(panel);
    map.events.register("zoomend", map, zoomChanged);
    var v1="<?php echo $_SESSION['cnt'] ?>";

    if (v1==1) {
        file_name="<?php echo $_SESSION['csvfile'] ?>";
        var tmpurl;
        if(file_name!=undefined)
            tmpurl="./uploads/"+file_name;

        $.ajax({
            type: "GET",
            url: tmpurl,
            dataType: "text",
            success: function(data) {
                allTextLines = data.split(/\r\n|\n/);
                headers = allTextLines[0].split(',');
                var longs=[];
                var lats=[];
                var data;
                var px, py;
                for(var i=1;i<allTextLines.length-1;i++){
                    data = allTextLines[i].split(',');
                    px = parseFloat(data[1]);
                    py = parseFloat(data[2]);
                    longs.push(px);
                    lats.push(py);
                }
            }
        });
    }
```

```
    }
}
```

Interactive Controls

Side Panel

Listing B.4: Side Panel design

```
<select id="chart_type" class="form-control">
    <option value="line">line</option>
    <option value="bar">bar</option>
    <option value="pie">pie</option>
    <option value="column">column</option>
</select>
<button href="#attrModal" id="openBtn" data-toggle="modal" style="
    margin-top:13px;" class="btn btn-info">Select attributes</button>
     <br/>
<button href="#aggModal" id="openBtn" data-toggle="modal" style="
    margin-top:13px;" class="btn btn-info">Aggregation function</
    button> <br/> <br/>
<button id="openBtn" data-toggle="modal" class="btn btn-primary"
    onclick="toggleLayer()">Toggle basemap</button> <br/> <br/>
<button id="openBtn" data-toggle="modal" class="btn btn-success"
    onclick="zoomPreProcess()">Run</button>
<div id="box">
    <button id="openBtn" data-toggle="modal" class="btn btn-info"
       onclick="reset()"><b>Reset</b></button> <br/> <br/>
    <b> Zoom: </b> <input type="text" id="chart" name="chart" value
       ="100" size="4">%        
    <button id="openBtn" data-toggle="modal" class="btn btn-info"
       onclick="chart()"><b>Submit</b></button><br/>
</div>
```

Chart Type

Listing B.5: Chart Type Selection methods

```
function changeChartType(charttype){
   options.chart.type=charttype;
   chart_type=charttype;
}

$('#chart_type').change(function () {
   charttype = $(this).find("option:selected").text();
   changeChartType(charttype);
});
```

Select Attributes

Listing B.6: Attribute Selection methods

```
<div class="modal fade" id="attrModal">
   <div class="modal-dialog modal-xl">
      <div class="modal-content">
         <div class="modal-header">
            <div class="modal-body">
               <div class="row">
                  <div class="col-xs-3">
                     Field/s for aggregation
                     <select id="type_agg" class="form-control">
                        <option value="--Select One--">--Select One
                           --</option>
                        <option value="single">Single Field</option>
                        <option value="multiple">Multiple Fields</
                           option>
                     </select>
                  </div>
                  <div class="col-xs-5">Boundary files uploaded</br>
```

```
                <script>
                    var filecount="<?php echo $_SESSION['zipcount']
                        ?>";
                    if(filecount<5 && filecount>0)
                        filecount=filecount-1;
                    var v2;
                    if(filecount>0){
                        v2="<?php echo $_SESSION['file1'] ?>";
                        document.write(v2);document.write("<br>");
                    }
                    if(filecount>1){
                        v2="<?php echo $_SESSION['file2'] ?>";
                        document.write(v2);document.write("<br>");
                    }
                    if(filecount>2){
                        v2="<?php echo $_SESSION['file3'] ?>";
                        document.write(v2);document.write("<br>");
                    }
                    if(filecount>3){
                        v2="<?php echo $_SESSION['file4'] ?>";
                        document.write(v2);document.write("<br>");
                    }
                    if(filecount>4){
                        v2="<?php echo $_SESSION['file5'] ?>";
                        document.write(v2);
                    }
                </script>
            </div>
        </div>
    </div>
    <div id="toBeDone"> </div> </br>
```

```
            <div class="modal-footer">
               <button type="button" class="btn btn-default" data-
                  dismiss="modal">Close</button>
            </div>
         </div>
      </div>
   </div>
</div>
```

```
$('#type_agg').change(function () {
   var type = $(this).find("option:selected").text();
   if(type=="Single Field"){
      casename=1;
      getData();
   }
   else if(type=="Multiple Fields"){
      casename=2;
      getData();
   }
});
```

```
function getData(){
   file_name="<?php echo $_SESSION['csvfile'] ?>";
   var tmpurl;
   if(file_name!=undefined)
      tmpurl="./uploads/"+file_name;
   $.ajax({
      type: "GET",
      url: tmpurl,
      dataType: "text",
      success: function(data) {
```

```
         allTextLines = data.split(/\r\n|\n/);
         headers = allTextLines[0].split(',');
      }
   });
   if(casename==1)
      addInput();
   else if(casename==2)
      addInput2();
}
```

```
function addInput() {
   document.getElementById("toBeDone").innerHTML = "";
   var someDiv = document.getElementById('toBeDone');
   var sel = document.createElement('select');
   sel.name = 'drop1';
   sel.id = 'agg';
   var selectHTML = "";
   var tmp = allTextLines[1].split(',');
   for(i = 3; i < headers.length; i = i + 1) {
      if(!isNaN(tmp[i]) && tmp[i].toString().match(/^-?\d*(.\d+)?$/)
         ){
         headers[i]=headers[i].replace(/['"]+/g, '');
         if(headers[i])
            selectHTML += "<option value='" + i + "'>" + headers[i]
               + "</option>";
      }
   }
   sel.innerHTML = selectHTML;
   someDiv.appendChild(sel);
}
```

```
function addInput2(){
   document.getElementById("toBeDone").innerHTML = "";
   var tmp = allTextLines[1].split(',');
   for (var i = 3; i<headers.length; i++){
      n=tmp[i];
      headers[i]=headers[i].replace(/['"]+/g, '');
      if(!isNaN(tmp[i]) && tmp[i].toString().match(/^-?\d*(.\d+)?$/)
          && headers[i]){
         var checkbox = document.createElement('input');
         checkbox.type = "checkbox";
         checkbox.name = headers[i];
         checkbox.value = i;
         checkbox.id = i;
         var newlabel = document.createElement('label');
         newlabel.setAttribute("for", 'toBeDone');
         newlabel.innerHTML = headers[i];
         var someDiv = document.getElementById('toBeDone');
         someDiv.appendChild(checkbox);
         someDiv.appendChild(newlabel);
         someDiv.appendChild(document.createElement("br"));
      }
   }
}
```

Data Aggregation

Listing B.7: Data Aggregation methods

```
<div class="modal fade" id="aggModal">
   <div class="modal-dialog modal-xl">
      <div class="modal-content">
         <div class="modal-header">
            <div class="modal-body">
```

```
                Choose Aggregation function </br>
                <div class="row">
                   <div class="col-xs-4">
                      <select id="aggregation_function" class="form-
                         control">
                         <option value="sum">sum</option>
                         <option value="min">min</option>
                         <option value="max">max</option>
                      </select>
                   </div>
                   <div class="col-xs-4">
                      <div id="append"> </div>
                      </br>
                   </div>
                </div>
                <div class="modal-footer">
                   <button type="button" class="btn btn-default "
                      data-dismiss="modal">Close</button>
                </div>
             </div>
          </div>
       </div>
   </div>
</div>
```

```
$('#aggregation_function').change(function () {
   agg_func = $(this).find("option:selected").text();
   changeAggFunc(agg_func);
});
```

```
function changeAggFunc(aggfunc){
```

```
   agg_func=aggfunc;
}
```

Toggle basemap

Listing B.8: Toggle Layer method

```
function toggleLayer() {
   if (osmlayer.getVisibility() == true){
      osmlayer.setVisibility(false);
   } else {
      osmlayer.setVisibility(true);
   }
}
```

Zoom by Area and Shift left or right

Listing B.9: Zoom Charts by Area method

```
OpenLayers.Control.CustomNavToolbar = OpenLayers.Class(OpenLayers.
   Control.Panel,{
   initialize: function(options) {
      OpenLayers.Control.Panel.prototype.initialize.apply(this, [
         options]);
      this.addControls([
         new OpenLayers.Control.Navigation(),
         new OpenLayers.Control.ZoomBox({alwaysZoom:true})
      ]);
      this.displayClass = 'olControlNavToolbar'
   },
   draw: function() {
     var div = OpenLayers.Control.Panel.prototype.draw.apply(this,
        arguments);
     this.defaultControl = this.controls[0];
     return div;
```

```
   }
});
var panel = new OpenLayers.Control.CustomNavToolbar();
map.addControl(panel);
```

Reset

Listing B.10: Reset method

```
function reset(){
   var num = map.getNumLayers();
   for (var j=1; j<num; j++) {
      map.removeLayer( map.layers[1] );
   }
}
```

Zoom All Charts

Listing B.11: Zoom All Charts method

```
function ZoomAllCharts() {
   var newchartradius=document.getElementById("chart").value;
   var num = map.getNumLayers();
   var mLayers = map.layers;
   for (var k=0; k<mLayers.length; k++) {
      if(mLayers[k].name == "POI Markers"){
         map.removeLayer(mLayers[k]);
         pointLayer = new OpenLayers.Layer.Vector("POI Markers");
         var pointFeatures = [];
         chartradius=(50)*(newchartradius/100);
         for(var i=0;i<chartindexs[levelcurr-1];i++){
            var px, py;
            py = Math.round(latitudes[levelcurr-1][i]*100)/100;
            px = Math.round(longitudes[levelcurr-1][i]*100)/100;
            var lonlat = new OpenLayers.LonLat(px, py);
```

```
            lonlat.transform(new OpenLayers.Projection("EPSG:4326"),
                new OpenLayers.Projection("EPSG:900913"));
            var pointGeometry = new OpenLayers.Geometry.Point(lonlat
               .lon, lonlat.lat);
            var j=i+1;
            var chartname=cname+j+'.png';
            var chartpath='./phantomjs/'+chartname;
            var pointFeature = new OpenLayers.Feature.Vector(
               pointGeometry, null, {
               pointRadius: chartradius,
               externalGraphic: chartpath,
            });
            pointFeatures.push(pointFeature);
            pointLayer.addFeatures(pointFeatures);
         }
         map.addLayer(pointLayer);
      }
   }
}
```

Data Based Processing

Listing B.12: Data Processing methods

```
function zoomPreProcess(){
   map.removeLayer(inLayer);
   zip_count="<?php echo $_SESSION['zipcount'] ?>";
   var difference = Math.floor(14/zip_count);
   var d=0, leveldecided=1;
   if(zip_count<5 && zip_count>0)
      zip_count=zip_count-1;
   var difference = Math.floor(18/5);
   for(var cnt=2;cnt<=18;cnt++) {
```

```
      m.push(leveldecided);
      d+=1;
      if(d>difference && zip_count > leveldecided) {
         d=0;
         leveldecided+=1;
      }
   }
   getVal();
}
```

```
function getVal() {
   cols=[];
   if(casename==2) {
      var test2 = document.getElementById('toBeDone').
         getElementsByTagName("checkbox");
      $checkedCheckboxes = $("input:checkbox:checked");
      $checkedCheckboxes.each(function () {
         cols.push($(this).val());
      });
   }
   else if(casename==1) {
      var test = document.getElementById('toBeDone').
         getElementsByTagName("select");
      cols.push(test[0].options[test[0].selectedIndex].value);
   }
   var typecnt="<?php echo $_SESSION['zipcount'] ?>";
   getData2(1,typecnt);
}
```

```
function getData2(type,typecnt){
   var tmpurl="./uploads/out"+type+".csv";
```

```
$.ajax({
   type: "GET",
   url: tmpurl,
   dataType: "text",
   success: function(data) {
      if(type==1)
         model1 = data.split(/\r\n|\n/);
      if(type==2)
         model2 = data.split(/\r\n|\n/);
      if(type==3)
         model3 = data.split(/\r\n|\n/);
      if(type==4)
         model4 = data.split(/\r\n|\n/);
      if(type==5)
         model5 = data.split(/\r\n|\n/);
      type++;
      if(type<typecnt && type<=4)
         getData2(type,typecnt);
      else if(type<=typecnt && type==5)
         getData2(type,typecnt);
      else{
         if(agg_func==undefined)
            agg_func='sum';
         if(chart_type==undefined){
            chart_type='line';
            options.chart.type='line';
         }
         else
            options.chart.type=chart_type;
         options.xAxis.categories=[];
         options.series=[];
```

```
            var str;
            for(var i=0;i<cols.length;i++){
               var len1 = headers[cols[i]].length;
               str=headers[cols[i]].substring(1, len1-1);
               headers[cols[i]]=str;
            }
            if(zip_count>0)
               processData(model1,1);
            if(zip_count>1)
               processData(model2,2);
            if(zip_count>2)
               processData(model3,3);
            if(zip_count>3)
               processData(model4,4);
            if(zip_count>4)
               processData(model5,5);
            n2=model1.length-1;
            chartsize=parseFloat(model1[n2-2]);
            casename2=2;
            init(1);
         }
      }
   });
}
```

```
function processData(model, type) {
   options.chart.type=chart_type;
   n2=model.length-1;
   chartindex=0;
   options.chart.type=chart_type;
   chartids[type-1]=model[n2-1].split(',');
```

```
   charttotal=chartids[type-1].length;
   chartradius=50;
   doLoop(0, model, type);
   chartindexs.push(chartindex);
}
```

```
function doLoop(i, model, type) {
   if(i<charttotal){
      var polypoint=[];
      polypoint = model[chartids[type-1][i]].split(',');
      py=parseFloat(polypoint[0]);
      px=parseFloat(polypoint[1]);
      if(polypoint.length > 2) {
         var temp=[];
         var temp2=[];
         var series = {
            data: []
         };
         series.showInLegend=false;
         series.lineWidth=30;
         if(i==0){
            longitudes.push([]);
            latitudes.push([]);
         }
         longitudes[type-1].push(px);
         latitudes[type-1].push(py);
         for(var k2=2;k2<polypoint.length;k2++){
            var csvdata = allTextLines[polypoint[k2]].split(',');
            if (csvdata.length == headers.length) {
               for (var j=0; j<cols.length; j++) {
                  if(k2==2){
```

```
                temp2.push(parseFloat(csvdata[cols[j]]));
            }
            else{
                if(agg_func=='sum')
                    temp2[j]=temp2[j]+parseFloat(csvdata[cols[j
                        ]]);
                else if(agg_func=='max') temp2[j]=Math.max(
                    temp2[j],parseFloat(csvdata[cols[j]]));
                else if(agg_func=='min')
                    temp2[j]=Math.min(temp2[j],parseFloat(
                        csvdata[cols[j]]));
            }
        }
    }
}
for(var k3=0;k3<temp2.length;k3++) {
    var res = headers[cols[k3]];
    options.xAxis.categories[k3] = res;
    temp.push(res);
    temp.push(temp2[k3]);
    series.data.push(temp);
    temp=[];
}
chartindex++;
options.series.pop();
options.series.push(series);
var cname;
if(type==1)
    cname='a';
else if(type==2)
    cname='b';
```

```
        else if(type==3)
           cname='c';
        else if(type==4)
           cname='d';
        else if(type==5)
           cname='e';
        var chartn=cname+chartindex+'.png';
        $.ajax({
           type: 'post',
           async : false,
           data: {"params":options, "chartname":chartn},
           url: './phantomjs/generateChart.php',
           success: function (data) {
              i++;
              doLoop(i, model, type);
           }
        });
        options.series=[];
     }
  }
}
```

Data Visualization

Listing B.13: Data Visualization methods

```
function init(level){
   var pointFeatures = [];
   pointLayer = new OpenLayers.Layer.Vector("POI Markers");
   levelcurr=level;
   var cname;
   if(level==1)
      cname='a';
```

```
else if(level==2)
   cname='b';
else if(level==3)
   cname='c';
else if(level==4)
   cname='d';
else if(level==5)
   cname='e';
if(casename2==2) {
   var tmpurl = "./uploads/out"+level+".geojson";
   var myStyle = {
      fill: true,
      fillColor: "#66cc33",
      fillOpacity: 0.2,
      strokeWidth: 3,
      strokeColor: "#000000",
      strokeDashstyle: "dash",
   };
   geojsonlayer = new OpenLayers.Layer.Vector("GeoJSON", {
      strategies: [new OpenLayers.Strategy.Fixed()],
      style: myStyle,
      protocol: new OpenLayers.Protocol.HTTP({
         url: tmpurl,
         format: new OpenLayers.Format.GeoJSON()
      })
   });
}
for(var i=0;i<chartindexs[level-1];i++){
   var px, py;
   py = Math.round(latitudes[level-1][i]*100)/100;
   px = Math.round(longitudes[level-1][i]*100)/100;
```

```
      var lonlat = new OpenLayers.LonLat(px, py);
      lonlat.transform(new OpenLayers.Projection("EPSG:4326"), new
         OpenLayers.Projection("EPSG:900913"));
      var pointGeometry = new OpenLayers.Geometry.Point(lonlat.lon,
         lonlat.lat);
      var j=i+1;
      var chartname=cname+j+'.png';
      var chartpath='./phantomjs/'+chartname;
      var pointFeature = new OpenLayers.Feature.Vector(pointGeometry
         , null, {
         pointRadius: chartradius,
         externalGraphic: chartname,
      });
      pointFeatures.push(pointFeature);
      pointLayer.addFeatures(pointFeatures);
   }
   map.addLayer(pointLayer);
   %var inlon = Math.round(latitudes[0]*100)/100;
   %var inlat = Math.round(longitudes[0]*100)/100;
}
```

```
function zoomChanged(){
   var zoom = map.getZoom();
   var leveldecided=m[zoom-2];
   if(leveldecided==levelcurr){
      casename2=1;
   }
   else{
      map.removeLayer(geojsonlayer);
      map.removeLayer(pointLayer);
      casename2=2;
```

```
      levelcurr=leveldecided;
      init(levelcurr);
   }
}
```

Appendix C

Additional Figures

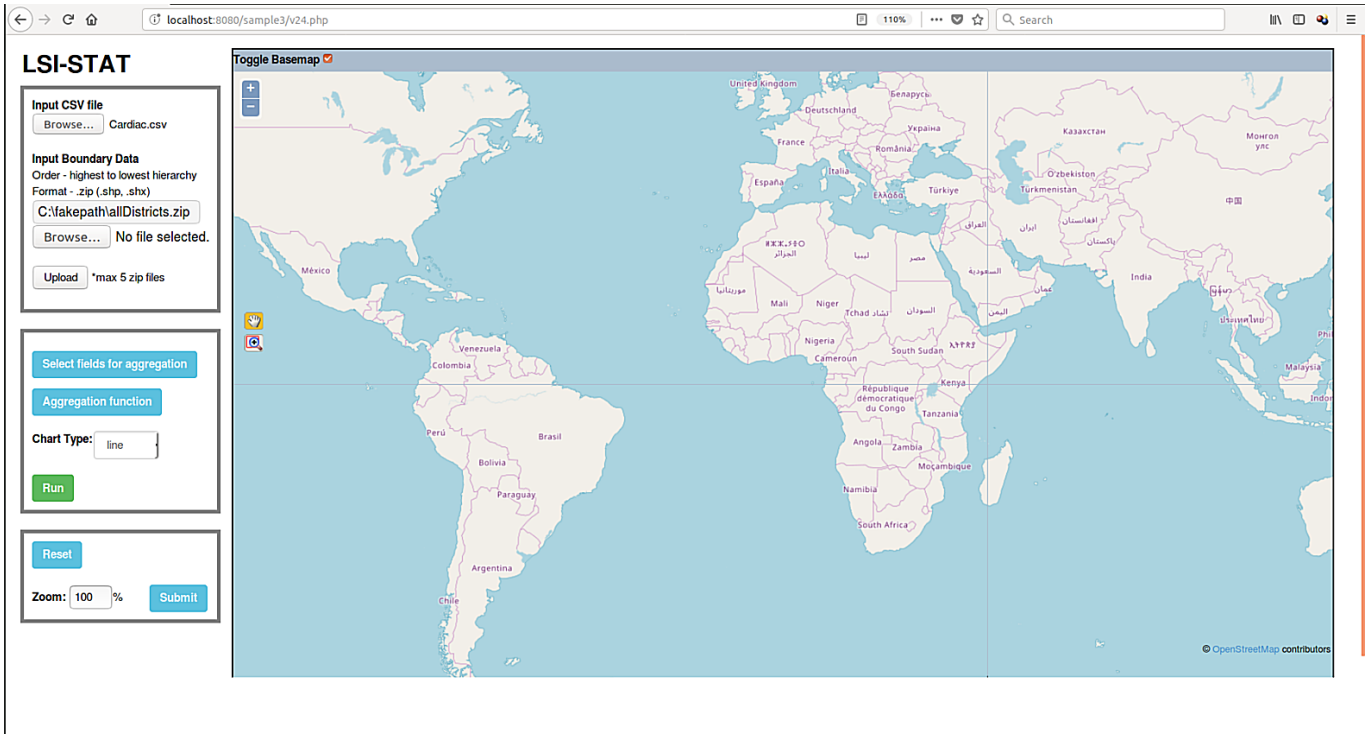

Figure C.1: Data Upload

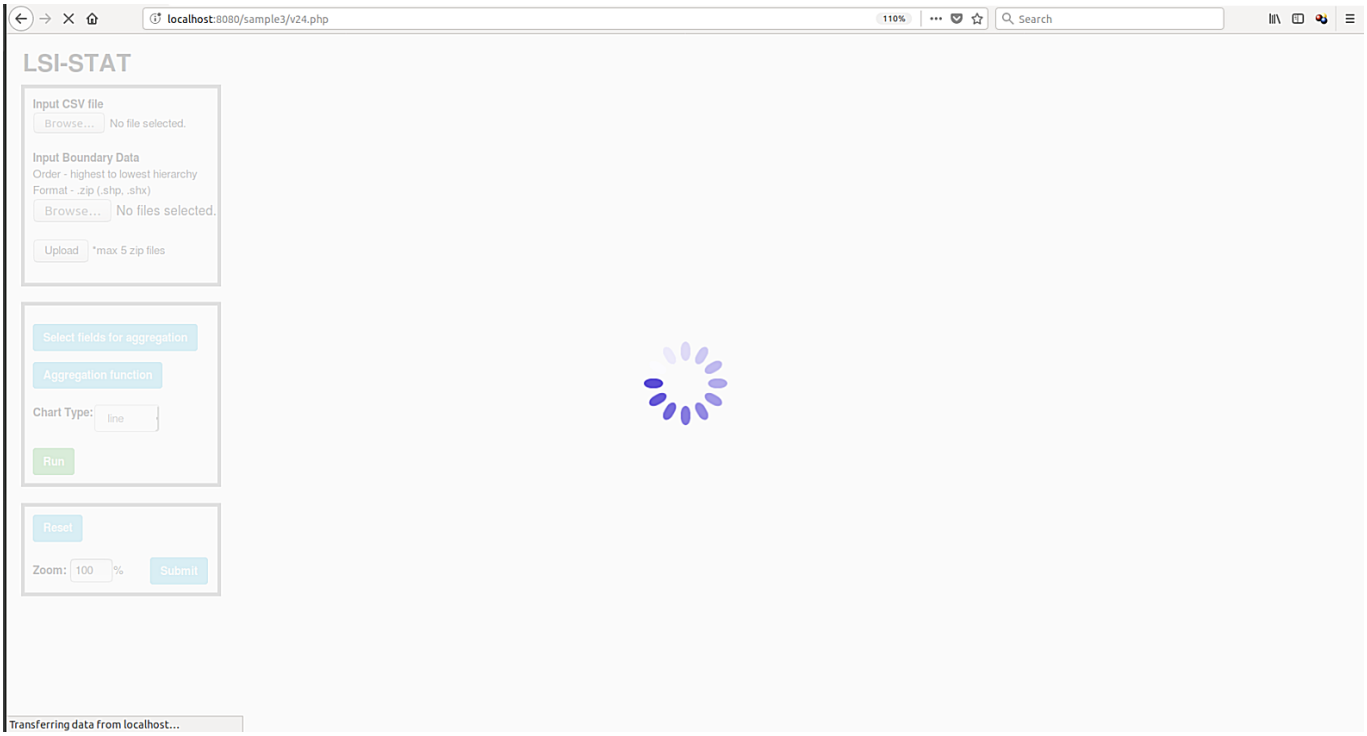

Figure C.2: Processing