Embedded Merge & Split: Visual Adjustment of Data Grouping

Major-Project Report submitted in partial fulfillment of the requirement for the award of degree of Bachelor of Technology

in

INFORMATION TECHNOLOGY

by

A.Supriya (17WH1A1216)

B.Divya Krupa (17WH1A1220)

BH.Deepika (17WH1A1221)

M.Sai Laya (17WH1A1237)

Under the esteemed guidance of

Mr.A.Rajashekar Reddy

Assistant Professor

Department of Information Technology

BVRIT HYDERABAD College of Engineering for Women

Rajiv Gandhi Nagar, Nizampet Road, Bachupally, Hyderabad – 500090

(Affiliated to Jawaharlal Nehru Technological University Hyderabad)

(NAAC 'A' Grade & NBA Accredited- ECE, EEE, CSE & IT)

June, 2021

DECLARATION

We hereby declare that the work presented in this project entitled "**Embedded Merge & Split: Visual Adjustment of Data Grouping**" submitted towards completion of the project in IV year II sem of B.Tech IT at "BVRIT HYDERABAD College of Engineering for Women", Hyderabad is an authentic record of our original work carried out under the esteem guidance of ***Mr.A.Rajashekar Reddy*, Assistant Professor**, IT Department.

A. Supriya
(17WH1A1216)

B.Divya Krupa
(17WH1A1220)

BH.Deepika
(17WH1A1221)

M.Sai Laya
(17WH1A1237)

BVRIT HYDERABAD

College of Engineering for Women

Rajiv Gandhi Nagar, Nizampet Road, Bachupally, Hyderabad – 500090

(Affiliated to Jawaharlal Nehru Technological University Hyderabad)

(NAAC 'A' Grade & NBA Accredited- ECE, EEE, CSE & IT)

CERTIFICATE

This is to certify that the work entitled **" Embedded Merge & Split: Visual Adjustment of Data Grouping"** is being submitted by **Ms.A.Supriya (17WH1A1216),Ms.B.DivyaKrupa(17WH1A1220),Ms.Bh.Deepika(17WH1A1221) ,Ms.M.SaiLaya (17WH1A1237)**. The Major Project Report Submitted in fulfillment of the curriculum, Bachelor of Technology in Information Technology affiliated to the **JAWAHARLAL NEHRU TECHNOLOGICAL UNIVERSITY HYDERABAD** is a record of bonafide work carried out by his under my guidance and supervision.

The results embodied in the project work have not been submitted to any other university or institute for the award of any degree or diploma

Mr. A.Rajashekar Reddy
Assistant Professor
Department of IT

Dr. Aruna Rao SL
Professor& HoD
Department of IT

External Examiner

ACKNOWLEDGEMENT

We would like to express our sincere thanks to **Dr. K. V. N. Sunitha, Principal, BVRIT HYDERABAD** for providing the working facilities in the college.

Our sincere thanks and gratitude to **Dr. Aruna Rao S L, Professor & Head**, **Department of IT, BVRIT HYDERABAD** for all the timely support and valuable suggestions during the period of our project.

We are extremely thankful and indebted to our internal guide,**Mr.A.Rajashekar Reddy, Assistant Professor, Department of IT, BVRIT HYDERABAD** for his constant guidance, encouragement and moral support throughout the project.

Finally, We would also like to thank our Project Coordinators **Dr. P.Kayal, Associate Professor, Dr.P.S.Latha K, Associate Professor,** all the faculty and staff of the IT Department who helped us directly or indirectly, parents and friends for their cooperation in completing the project work.

A. Supriya
(17WH1A1216)

B.Divya Krupa
(17WH1A1220)

BH.Deepika
(17WH1A1221)

M.Sai Laya
(17WH1A1237)

ABSTRACT

Data grouping is among the most frequently used operations in data visualization. It is the process through which relevant information is gathered, simplified, and expressed in summary form.Many popular visualization tools support the automatic grouping of data (e.g., dividing up a numerical variable into bins). Although grouping plays a pivotal role in supporting data exploration, further adjustment and customization of auto-generated grouping criteria are non-trivial. Such adjustments are currently performed either programmatically or through menus and dialogues which require specific parameter adjustments over several steps. In response, we introduce Embedded Merge & Split (EMS), a new interaction technique for direct adjustment of data grouping criteria. We demonstrate how the EMS technique can be designed to directly manipulate width and position in bar charts and histograms of data grouping criteria. We also offer a set of design guidelines for supporting EMS. Finally, we present user studies, providing initial evidence that EMS can significantly reduce interaction time compared to WIMP-based techniques and was subjectively preferred by participants. Index Terms of this project are Data Visualization, Direct Manipulation, Embedded Merge & Split, Data Grouping, Embedded Interaction.

LIST OF FIGURES

LIST OF TABLE

LIST OF ABBREVIATIONS

Term/Abbreviation	Definition
EMS	Embedded merge and split
WIMP	Windows,Icons,Mouse,Pull-down menus
NYC	New York City
SAS	Statistical Analysis System

CONTENT

1. INTRODUCTION

1.1 RELATED WORK

Data exploration is one of the most crucial steps to get a better understanding of the Dataset. This step is used to identify the patterns and interesting insights into the data. To understand the Dataset, an overview of the distribution of different attributes within that Dataset is essential. One of the most common approaches to getting this overview is through Data Grouping. Data Grouping enables the users to get a glimpse of identifying underlying patterns and relations among the data. With advancements in technology, data is enormously increasing in size every minute, which becomes a tedious task for a person to analyze and explore the data.

The primary goal of this paper lies in grouping large amounts of data dynamically that does not involve manual work. In traditional visualization tools like Tableau, we perform these types of activities using a menu-based interaction in a series of actions. For example, the user would open a menu, and then select the desired grouping characteristics, and the visualization would update accordingly. This approach is not sufficient because the analyst has to go through a series of steps to achieve the desired level of grouping.

To analyze the entire data, the user might have to change the grouping characteristics and attributes. Such requirements become the root behind the system "Embedded Merge and Split". The main intention of this project was to enable dynamic interactivity during data grouping operations.

1.2 PROBLEM DESCRIPTION

The embedded merge & split offers embedded interactivity directly with the charts to visualize changes instantly instead of using a menu-based design, which can incur extra execution and cognitive costs, especially as the set of available operations increases. With data grouping in mind, we as a group decided to work on data sets that could be over-viewed using the most common data grouping graphs like bar charts, histograms, etc.

In order to achieve effective results that help the user to find the summaries and reveal the patterns within data, we proposed an extension to the system that helps in adding these interactions to another visualization called a matrix plot. Such communication might prove useful as a way to incorporate a comparative analysis of the overall data as illustrated.

1.3 AIM OF THE PROJECT

The main goal of this project is to build an interactive interface for user's which makes user's complex tasks simple, In this interface the user's dataset (csv file) is been visualized into different patterns based on the given information in the file and best part is that user can perform different operations on those visualised patterns and conclude differents solutions and patterns from it.

1.3.1 DATASET DESCRIPTION

In our project we took four datasets: the house price prediction Dataset, the New York City AirBNB Open Dataset, Border Crossing Entry Dataset, Suicides in India for visualization. The first dataset describes houses across India with the amount of square feet, bhk number, posted_by status etc. , the second describes the listing activity and metrics for New York City and the third dataset provides summary statistics at the port level for inbound crossings collected by US Customs and Border Protection and fourth represents the suicides and the reasons behind them along with the number of gender across various states of India.

1.3.2 .FILE DESCRIPTION

House Price Prediction DataSet:

Our first dataset is the House price prediction dataset. This dataset consists of the information about house prices. Accurately predicting house prices can be a daunting task. The buyers are just not concerned about the size(square feet) of the house and there are various other factors that play a key role to decide the price of a house/property. It can be extremely difficult to figure out the right set of attributes that are contributing to understanding the buyer's behavior as such. This dataset has been collected across various property aggregators across India.This helps the users to predict the house price. This dataset set consists of 11 rows and 68720 rows.

Column Name	Description
posted_by	category marking who has listed the property
under_construction	under construction or not
rera	rera approved or not
bhk_no	number of rooms
bhk or rk	type of property

square_ft	total area of the house in square feet
ready to move	category marking ready to move or not
resale	category marking resale or not
address	address of the property
longitude	longitude of the property
latitude	latitude of the property

Table 1.1 Dataset-1 File description

New York City Airbnb Open DataSet:

Our second dataset was the New York City AirBNB open dataset from kaggle. In this dataset, there are 36.5Million Rows and 16 columns. This dataset describes the listing activity and metrics for New York City. This dataset includes all the necessary information about hosts, geographical availability reviews, and other parameters. The current system visualizes the categorical attributes in terms of a bar chart and a matrix plot, whereas all the numerical data is visualized using a histogram. The system also visualizes data in a choropleth map that shows the number of reviews per neighborhood group in NYC. It also provides a filter that lets the user visualize various attributes as per his requirement and screen according to values within a quality. This system allows the user to discover interesting patterns and new trends by exploring the data graphically using the merge and split interactions making his analysis easier.

Column Name	**Description**
id	listing ID
name	name of the listing
host_id	ID of the host
host_name	name of the host
neighbourhood_group	location
neighbourhood	area
latitude	latitude coordinates
longitude	longitude coordinates
room_type	listing space type
price	price in dollars

minimum_nights	amount of nights minimum
number_of_reviews	number of reviews
last_review	latest review
reviews_per_month	number of reviews per month
calculated_host_listing	amount of listing per host
availability_365	number of days listing is available for booking

Table 1.2 Dataset-2 File description

Border Crossing Entry DataSet:

Our third dataset was the US Border crossing entry dataset. This data set provides summary statistics at the port level for inbound crossings collected by US Customs and Border Protection. The data available is for various types of vehicles, including pedestrians. The current system tries to visualize the frequency of border crossing concerning multiple attributes such as state, mode of passing, border, etc. The system can find its application to reveal the trends and patterns followed in border crossings that could help him identify and resolve the issue. It makes it easier for the user to understand various patterns between attributes by filtering data concerning port code and analyzing the graphs using merging and splitting. Changing the number of bins within the histogram finds its application in such a case when data within a range is comparatively lower.

Column Name	**Description**
port name	name of cbp port of entry
state	name of the state
port code	cbp port code
border	US-Canada border or US- Mexico border
date	specific date
measure	conveyances,containers,passengers, pedestrians
value	count
location	latitude and longitude location

Table 1.3 Dataset-3 File description

Suicides in India Dataset:

Our fourth data set contains yearly suicide detail of all the states/u.t of India by various parameters from 2001 to 2012. It has a granularity every year. Accurately predicting the suicide rate is a daunting task. Our dataset has data which includes many attributes like type code which states the reason. It even contains details about gender which determines if the person is ofmale or female. It has an age group attribute which depicts which age group is more prone to it. And finally has a total count of suicides committed.

Column Name	Description
state	state of the person
year	year to tragedy
type_code	reason of death
type	cause of death
gender	gender of the person
age_group	age group of the person
total	total number of person dead
latitude	location coordinates of state
longitude	location coordinates of state

Table 1.4 Dataset-4 File description

2. LITERATURE SURVEY

2.1 EXISTING SYSTEM

Many existing visualization tools such as Tableau and MS Excel support creation and presentation of grouped data. In response to user specifications, these tools perform data grouping and present the data. For example, in Tableau, we can specify our interest in having a visualization that has patients' ages on the x-axis and their population on the y-axis. In response, the system automatically groups patients' ages into a smaller number of age bins and represents the grouped data using a histogram visualization.

The system automatically performs data grouping based on statistical properties and inherent classes of data and presents it. During visual data exploration processes, users constantly need to adjust the predefined data groupings created by visualization tools based on their evolving needs and interests. However, a majority of the existing visualization tools such as Tableau and Spotfire require users to adjust data groupings by going through a broad set of menus provided on control panels (WIMP-based technique).

The WIMP-based technique can incur extra execution and cognitive costs especially as the number of available operations increases. Interfaces with a large number of execution steps may deter efficient use by inducing additional cognitive loads. Advanced adjustment of data grouping is only supported by highly specialized data analysis softwares (e.g., SAS) and visualization authoring tools such as Lyra . However, utilizing this class of tools typically requires advanced knowledge of visualization design and/or programming.

2.2 PROPOSED SYSTEM

We have proposed a system such that it has various panels in it. . In response to users actions, the system recognizes the users requirements and reconstructs the views as per the new specifications provided by the user. The proposed system lets the user split the merged categorical data and numeric data and visualize the bars individually. By simple drag functionality it lets the user update the number of bins and divide the data accordingly. The system is useful in applications where the data is massive, and the subset of data required for each requirement changes frequently. The current system visualizes data in various forms that include

1. Bar Graph
2. Histogram
3. Map
4. Line Graph
5. Matrix Plot

2.3 RESEARCH

AUTHOR	TITLE	DESCRIPTION	LIMITATIONS
Matthew N. O. Sadiku, Adebowale E. Shadare, Sarhan M. Musa and Cajetan M. Akujuobi	Data Visualization	Presenting data in graphical or pictorial form which makes the information easy to understand	The representations doesn't involve any user interactions
Inseok Ko, Hyejung Chang	Interactive Visualization of Healthcare Data Using Tableau	Procedure for the interactive visualization and analysis of healthcare data using Tableau as a business intelligence tool	supported using the Window/Icon/Menu/Pointer (WIMP) model which requires specific parameter adjustments over several steps.
Robert F. Eracher, Philip C. Chen, Jonathan C. Roberts, Craig M. Wittenbrink	Visual Data Exploration and Analysis VII	Easy to visually explore extremely large datasets. Loaded with both Infographics and Visual Explorer functionality	Users constantly need to adjust the predefined data groupings created by visualization tools based on their evolving needs and interests.

Table 2.1 Literature survey

3. SYSTEM ANALYSIS DESIGN

3.1 ARCHITECTURE DESIGN

The figure 3.1 represents the architecture diagram of this project. Firstly we selected the data which is suitable for this project then we loaded the collected data into the interface then the data has been analysed and characterized according to the information present in data. This analysed data is required output which is represented in the form of bar graphs,histograms,line graphs and map view and the best part is even the users can perform the visualizations like merge,split and filtering on these data interpretations to extract the users required patterns.

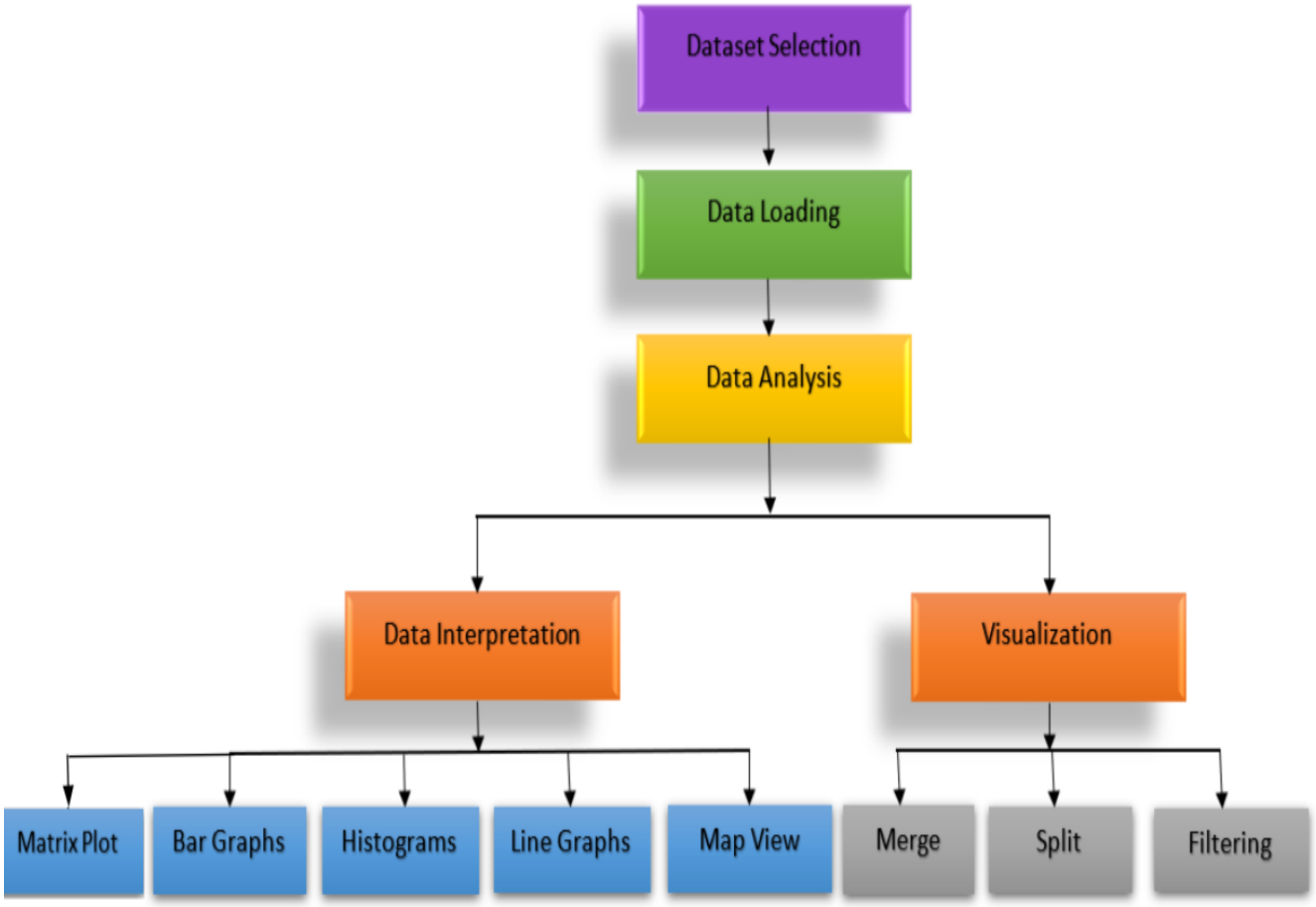

Figure 3.1 : System Design

3.2 USE CASE DIAGRAM

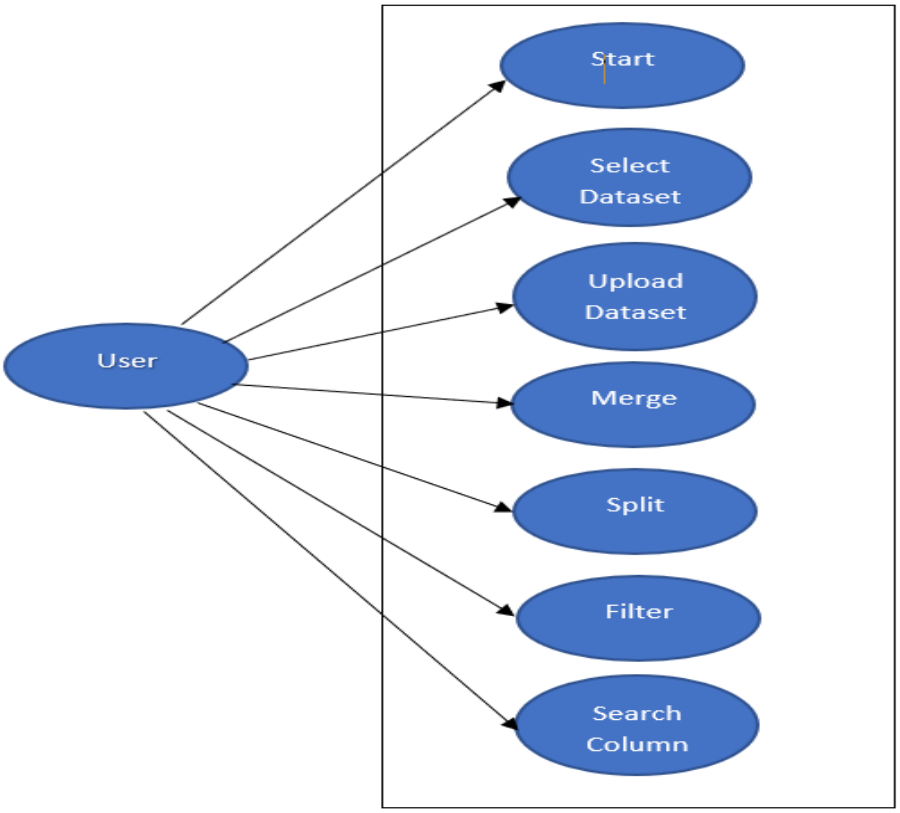

Figure 3.2 use case diagram

3.3 SEQUENCE DIAGRAM

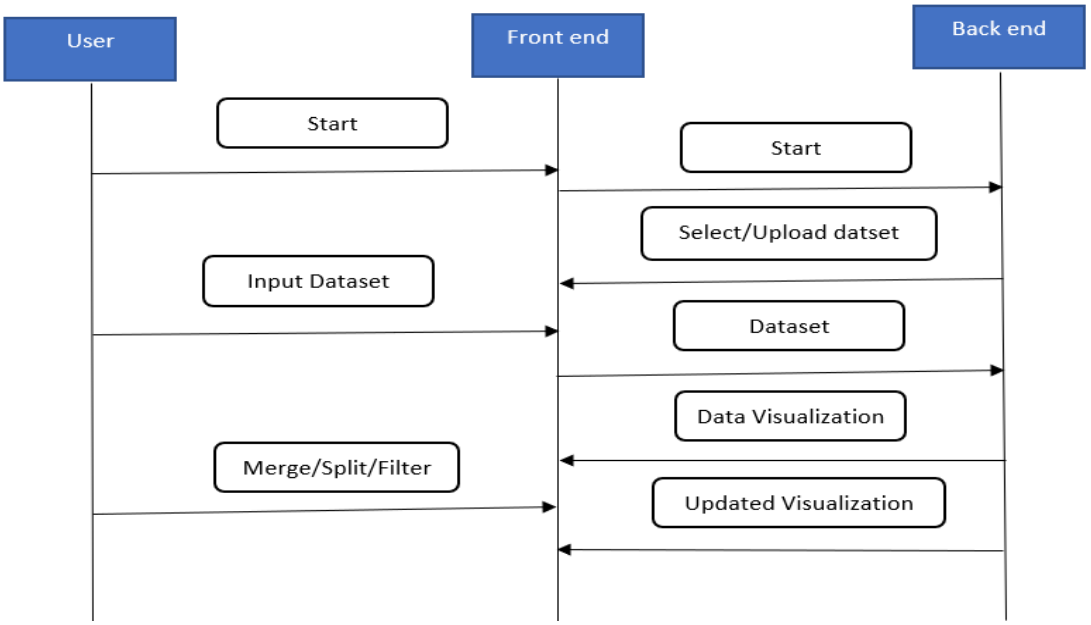

Figure 3.3 sequence diagram

3.4 REQUIREMENT SPECIFICATION

To be used efficiently, all computer software needs certain hardware components or other software resources to be present on a computer. These prerequisites are known as system requirements and are often used as a guideline as opposed to an absolute rule. Most software defines two sets of system requirements, minimum and recommended.

3.4.1. HARDWARE REQUIREMENTS

The hardware requirements list is often accompanied by a hardware compatibility list(HCL), especially in case of operating systems. An HCL lists tested, compatible, and sometimes incompatible hardware devices for a particular operating system or application. The following subsection discusses the various aspects of hardware requirements.

- ➢ Operating System - Windows 8 or higher version is required
- ➢ Hard Driver - Minimum 32Gb, Recommended 64Gb
- ➢ Processor - Intel core i5 and 2.50 GHz processor
- ➢ Physical Memory - Minimum 8Gb RAM is required

3.4.2. SOFTWARE REQUIREMENTS

The software requirements are descriptions of features and functionalities of the target system. Requirements convey the expectations of users from the software project.

- ➢ Operating system : Windows XP/10
- ➢ Coding Language : Python , Javascript & Html
- ➢ IDE : Visual studios or Pycharm
- ➢ Cloud service : Google cloud server

3.4.2.1 PYTHON

Python is an interpreted, object-oriented, high-level programming language with dynamic semantics. Its high-level built in data structures, combined with dynamic typing and dynamic binding, make it very attractive for Rapid Application Development, as well as for use as a scripting or glue language to connect existing components together. Python's simple, easy to learn syntax emphasizes readability and therefore reduces the cost of program maintenance. Python supports modules and packages, which encourages program modularity and code reuse. The Python interpreter and the extensive standard library are available in source or binary form without charge for all major platforms, and can be freely distributed. Since there is no compilation step, the edit-test-debug cycle is incredibly fast.

Debugging Python programs is easy: a bug or bad input will never cause a segmentation fault. Instead, when the interpreter discovers an error, it raises an exception. When the program doesn't catch the exception, the interpreter prints a stack trace. A source level debugger allows inspection of local and global variables, evaluation of arbitrary expressions, setting breakpoints, stepping through the code a line at a time, and so on. The debugger is written in Python itself, testifying to Python's introspective power. On the other hand, often the quickest way to debug a program is to add a few print statements to the source: the fast edit-test-debug cycle makes this simple approach very effective.

3.4.2.2 JAVASCRIPT

JavaScript(often shortened to JS) is a lightweight, interpreted, object-oriented language with first-class functions, and is best known as the scripting language for Web pages, but it's used in many non-browser environments as well. It is a prototype-based, multi-paradigm scripting language that is dynamic, and supports object-oriented, imperative, and functional programming styles.

JavaScript runs on the client side of the web, which can be used to design / program how the web pages behave on the occurrence of an event. JavaScript is an easy to learn and also powerful scripting language, widely used for controlling web page behavior.Contrary to popular misconception, JavaScript is *not* "Interpreted Java". In a nutshell, JavaScript is a dynamic scripting language supporting prototype based object construction. The basic syntax is intentionally similar to both Java and C++ to reduce the number of new concepts required to learn.

JavaScript can function as both a procedural and an object oriented language. Objects are created programmatically in JavaScript, by attaching methods and properties to otherwise empty objects at run time, as opposed to the syntactic class definitions common in compiled languages like C++ and Java. Once an object has been constructed it can be used as a blueprint (or prototype) for creating similar objects.

3.4.2.3 HTML

The HyperText Markup Language, or HTML is the standard markup language for documents designed to be displayed in a web browser. It can be assisted by technologies such as Cascading Style Sheets (CSS) and scripting languages such as JavaScript.

Web browsers receive HTML documents from a web server or from local storage and render the documents into multimedia web pages. HTML describes the structure of a web page semantically and

originally included cues for the appearance of the document.HTML elements are the building blocks of HTML pages. With HTML constructs, images and other objects such as interactive forms may be embedded into the rendered page.

HTML provides a means to create structured documents by denoting structural semantics for text such as headings, paragraphs, lists, links, quotes and other items. HTML elements are delineated by *tags*, written using angle brackets. Tags such as <img /> and <input /> directly introduce content into the page. Other tags such as <p> surround and provide information about document text and may include other tags as sub-elements. Browsers do not display the HTML tags, but use them to interpret the content of the page.

HTML can embed programs written in a scripting language such as JavaScript, which affects the behavior and content of web pages. Inclusion of CSS defines the look and layout of content. The World Wide Web Consortium (W3C), former maintainer of the HTML and current maintainer of the CSS standards, has encouraged the use of CSS over explicit presentational HTML since 1997.

3.4.2.4 CSS

Cascading Style Sheets (CSS) is a style sheet language used for describing the presentation of a document written in a markup language such as HTML.CSS is a cornerstone technology of the World Wide Web, alongside HTML and JavaScript.CSS is designed to enable the separation of presentation and content, including layout, colors, and fonts.This separation can improve content accessibility, provide more flexibility and control in the specification of presentation characteristics, enable multiple web pages to share formatting by specifying the relevant CSS in a separate .css file which reduces complexity and repetition in the structural content as well as enabling the .css file to be cached to improve the page load speed between the pages that share the file and its formatting.Separation of formatting and content also makes it possible to present the same markup page in different styles for different rendering methods, such as on-screen, in print, by voice (via speech-based browser or screen reader), and on Braille-based tactile devices. CSS also has rules for alternate formatting if the content is accessed on a mobile device.

3.4.2.5 SVG

SVG stands for Scalable Vector Graphics. SVG is an XML-based vector graphics format. It provides options to draw different shapes such as Lines, Rectangles, Circles, Ellipses, etc. Hence, designing visualizations with SVG gives you more power and flexibility.

Some SVG characteristics are listed below:

- SVG is text-based, and it is an image format that is vector-based.
- SVG is the same as the HTML structure.
- It can be illustrated as the **DOM (Document Object Model)**.
- The properties of SVG can be described as attributes.
- SVG must contain absolute positions that are relative to an origin position (0, 0).
- SVG may be added as is inside any HTML document.

3.4.2.6. IDE

PYCHARM

PyCharm is an integrated development environment (IDE) used in computer programming, specifically for the Python language. It is developed by the Czech company JetBrains (formerly known as IntelliJ).It provides code analysis, a graphical debugger, an integrated unit tester, integration with version control systems (VCSes), and supports web development with Django as well as data science with Anaconda.PyCharm is cross-platform, with Windows, macOS and Linux versions. The Community Edition is released under the Apache License, and there is also Professional Edition with extra features – released under a proprietary license.

VISUAL STUDIOS

Microsoft Visual Studio is an integrated development environment (IDE) from Microsoft. It is used to develop computer programs, as well as websites, web apps, web services and mobile apps. Visual Studio uses Microsoft software development platforms such as Windows API, Windows Forms, Windows Presentation Foundation, Windows Store and Microsoft Silverlight. It can produce both native code and managed code.Visual Studio includes a code editor supporting IntelliSense (the code completion component) as well as code refactoring. The integrated debugger works both as a source-level debugger and a machine-level debugger. Other built-in tools include a code profiler, designer for building GUI applications, web designer, class designer, and database schema designer.

It accepts plug-ins that expand the functionality at almost every level—including adding support for source control systems (like Subversion and Git) and adding new tool sets like editors and visual

designers for domain-specific languages or toolsets for other aspects of the software development lifecycle (like the Azure DevOps client: Team Explorer).Visual Studio supports 36 different programming languages and allows the code editor and debugger to support (to varying degrees) nearly any programming language, provided a language-specific service exists. Built-in languages include C,C++, C++/CLI, Visual Basic .NET, C#, F#, JavaScript, TypeScript, XML, XSLT, HTML, and CSS. Support for other languages such as Python, Ruby, Node.js, and M among others is available via plug-ins. Java (and J#) were supported in the past.The most basic edition of Visual Studio, the Community edition, is available free of charge. The slogan for Visual Studio Community edition is "Free, fully-featured IDE for students, open-source and individual developers".

3.5. LIBRARIES

In order to visualize the datasets, we have used the Javascript library. Below are the few packages and libraries that are needed to be installed and imported to run our project figure 1.3 is the screen shot of installed packages.

D3 Library:

D3.js (also known as **D3**, short for **Data-Driven Documents**) is a JavaScript library for producing dynamic, interactive data visualizations in web browsers. It makes use of Scalable Vector Graphics (SVG), HTML5, and Cascading Style Sheets (CSS) standards. It is the successor to the earlier Protovis framework.Its development was noted in 2011,as version 2.0.0 was released in August 2011. The D3.js library uses pre-built functions to select elements, create SVG objects, style them, or add transitions, dynamic effects or tooltips to them. These objects can also be styled using CSS. Large datasets can be bound to SVG objects using D3.js functions to generate text/graphic charts and diagrams. The data can be in various formats such as JSON, comma-separated values (CSV) or geoJSON, but, if required, JavaScript functions can be written to read other data formats.

Figure 3.4 D3 Library

4. IMPLEMENTATION

In this project, we implemented Embedded Merge & Split, an embedded interaction technique that lets users adjust data grouping criteria by directly manipulating the visualizations and filtering values through the main control panel. We replicated EMS for bar charts and histograms, as presented in our reference paper. We extended the implementation of this visualization to support matrix plots, extend merge and split in bar charts to allow separation of the stacks irrespective of them being on top of the stack, which was not the case in the original paper. The visualization was successfully implemented and tested with two data sets. The system was easy to use, intuitive, and the extensions to the system added useful functionalities. The front-end UI Avant-Grade was developed in a sophisticated manner, and the system works efficiently and smoothly with all types of users.We took the proposed EMS interactions a step further, and we believe that much more progress can be made in this research area that helps the users with easier accessibility to the system and adds more detail to enormous data.

4.1. MODULES

4.1.1.BAR GRAPH

Bar charts are used to compare individual values (e.g., frequency) between several groups .Bar charts use size to encode scalar values in height and position to encode grouping information. Typically, bar charts are used for categorical variables and categories inherent in data are used for grouping values.

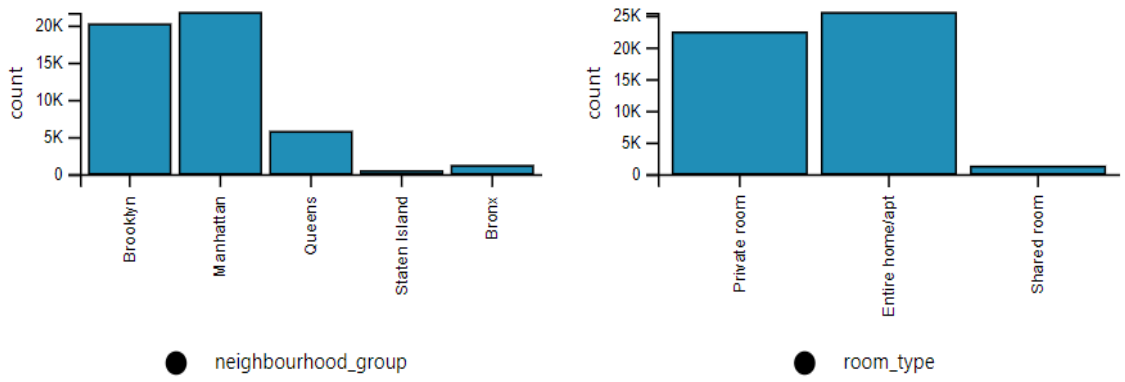

Figure 4.1 Bar Graphs

Merge – To merge two or more groups using the EMS technique, a user starts the operation by selecting a bar to be merged . Upon doing this, the selected bar will be highlighted to confirm the user's selection. Triangular icons appear in the middle of the bar to guide the user's interaction (G1). Selected bars (i.e., group) can be dragged using either left or right signifiers towards a target bar to be merged with (G2). While dragging the bar, the label at the top is updated continuously to indicate that the desired target is reached (G3). When the user drops the selected bar at the target, the system recomputes new grouping of values and then reconstructs and presents the bar chart. The combined bars will be shown as a stacked bar. Users can also select multiple bars and combine them with a target. See Figure 5-a for more details. Split – To split two groups that are merged into a bar, a user first selects the bar that needs to be split. The selected bar will be highlighted to confirm user selection, and signifiers appear in the middle of the bar to guide user interaction (G1). Users can drag the bar upward or downward to indicate their interest in splitting the bar (G2). When the user stops dragging, the system will reconstruct the bar chart, and a bar will be added to present the new grouping (G3). Figure 5.2illustrates the process of splitting bins in a bar chart using the EMS technique.

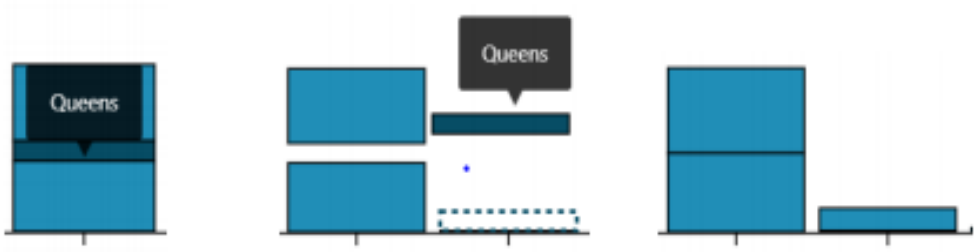

Figure 4.2 Bar Graph operations

4.1.1.1.Steps in bar charts

Step 1: Start with creating the SVG and defining the scales for our bar chart

Step 2: Let's load our data from the CSV file and add axes to the SVG.

Step 3: Next, we want to create bars corresponding to the data values.

Step 4: Add labels to the bar chart and tooltips for ease.

Step 5: Create a drag object that performs merge and split.

4.1.2.HISTOGRAMS

Histograms are one of the most commonly used visualizations for representing distribution. Histograms are suitable for investigating the shape of the distribution as well as the values. Each bar in the histogram represents a bin where the width of the bar typically shows the range of the values and the height of the bar shows the frequency (count). Merge – To merge two or more bins using the EMS technique, users first click on a bar to select it. In response to users' action, the selected bar becomes highlighted to visually confirm users' selection. Triangular icons on the outside of the selected bar signify that the bar can be extended to the left or right (G1). Additionally, upon selection of the bar, a label appears on the top of the selected bar to show its current range. A user can drag the triangular icons to left/right to alter (decrease/increase) the number of bins (G2). The system simultaneously recomputes and renders the histogram according to new specifications. The width of the altered bar will change in proportion to the new range (G3). Figure 5.2 illustrates the process of merging bins in a histogram using the EMS technique. Split – This action enables the users to divide a bin into smaller sub-bins. The implementation is very similar to that of the merge, with the exception that inner triangular icons are used to signify and alter the width of the bar.

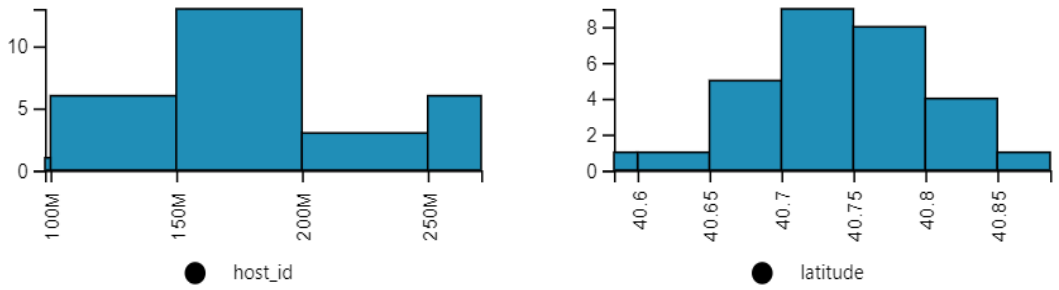

Figure 4.3 Histograms

Changing the number of bins – To change the number of bins using the EMS technique, a user draws a rubber-band rectangle to select the histogram. After the selection, the rectangle covers the entire histogram with two triangular icons signifying where and how to interact with the histogram (G1). Additionally, a label appears on the top of the selected histogram to show its current range. The user can drag the triangular icons to left/right to alter (decrease/increase) the number of bins (G2). The system continuously provides feedback about the current number of bins as the user drags the rubber-band to the left or right using a label placed on top (G3). When dragging is stopped, the

system computes the new bin ranges according to the number of bins and updates the histogram accordingly. See Figure 4 for more details.Changing the number of groups – In histograms, a numerical variable is grouped into a smaller number of bins. With numerical variables, grouping is subjective. For example, while one might decide to group the Age variable into 5 bins where the range of each bin is 10, another might group the variable into 10 bins where the range of each bin is 5. Thus, in histograms, we can change the number of bins by altering the range of each bin (the higher the number of bins, the smaller the range of each bin). On the other hand, grouping of categorical variables is usually performed based one the inherent categories that exist in data (e.g., Animal Types [Mammal, Bird, Reptiles]). Therefore, changing the number of groups for categorical variables is not supported by the current version of EMS.

4.1.2.1.Steps in Histograms

Step 1: Start with creating the SVG and defining the scales for our histogram

Step 2: Let's load our data from the CSV file and add axes to the SVG.

Step 3: Next, we want to create bars corresponding to the data ranges.

Step 4: Add labels to the histograms and tooltips for ease.

Step 5: Create a drag object that performs merge and split and updates the number of bins accordingly.

4.1.3. LINE GRAPH

The line graph visualizes data over time. It shows the information connected in some way. It is used to generalize the trends over some time.It is a type of chart used to show information that changes over time. We **plot line graphs** using several points connected by straight **lines**. We also call it a **line** chart. The **line graph** comprises two axes known as 'x' axis and 'y' axis. The horizontal axis is known as the x-axis.

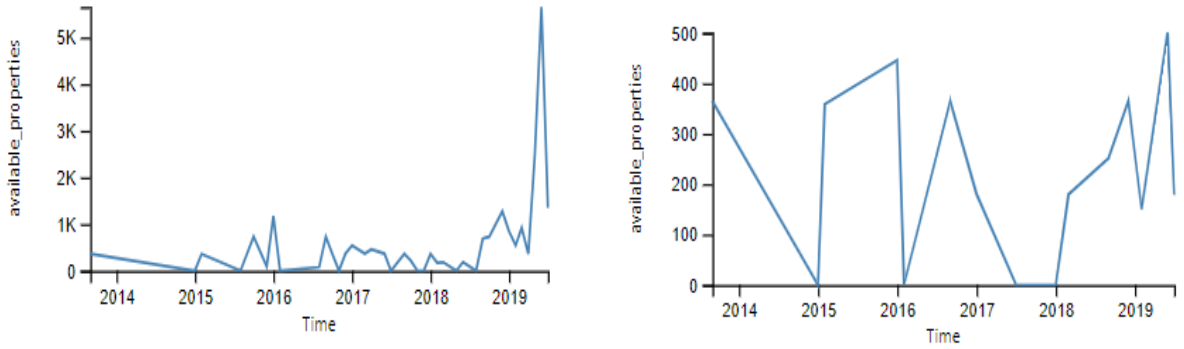

Figure 4.4 Line Graphs

4.1.3.1.Steps in Line graph

Step 1: Start with creating the SVG. It specifies the chart size and margins.

Step 2: Let's load our data from the CSV file and add axes to the SVG.

Step 3: Next, we need to form the line using path and d3.line utility.

Step 4: Add labels to the line graphs and tooltips for ease.

Step 5: Add a rectangle on top of the svg area that will track the mouse position using style("pointer-events","all").

Step 6: Then, we recovered the nearest x coordinate in the dataset using d3.bisector() function. Once we have the position we can use it to update the circle and text position on the chart.

4.1.4 MAP

This visualizes the aerial view of the corresponding area and produces the explicit details within the sub-area. The tool tips provide additional information that is useful to the user for his analysis.

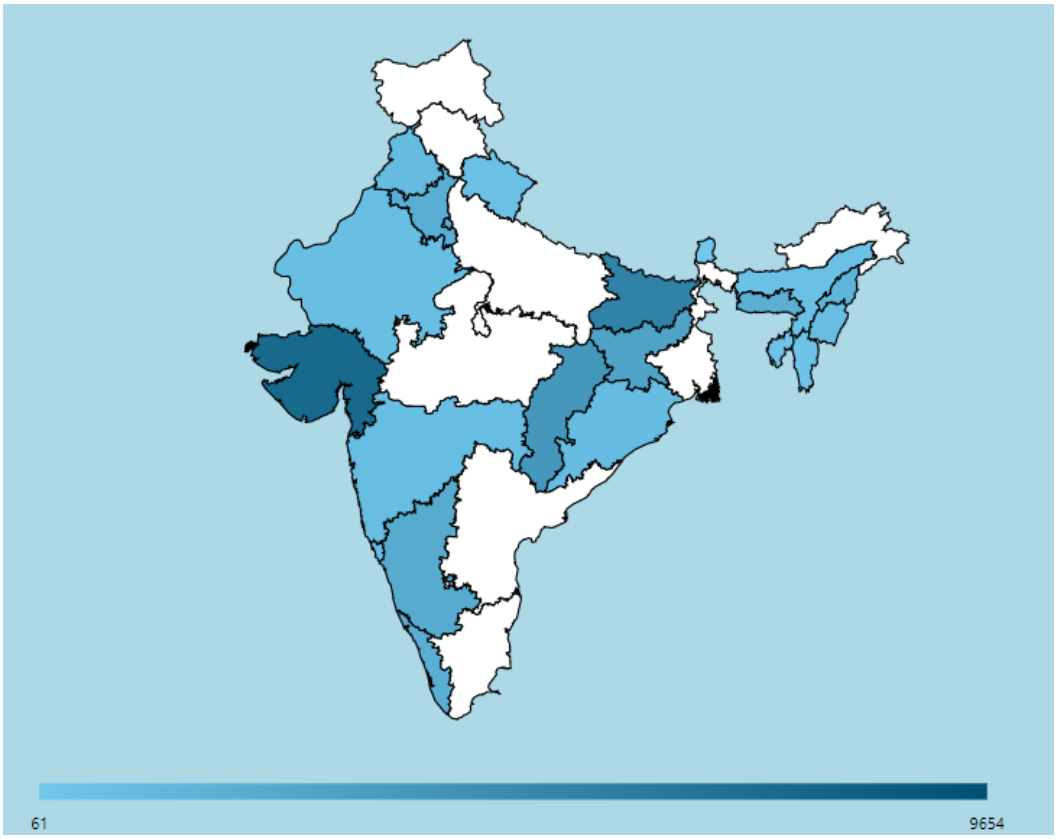

Figure 4.5 Map View

4.1.4.1.Steps in Map

Step 1: Start with creating the SVG.

Step 2: Here country boundaries are used. This data is loaded in json format and appended to the svg.

Step 3: Lets load our data from the csv file.

Step 4: A threshold color scale is used to depict the average of each sub part in the map.

Step 5: Define the map settings and add tooltips that display the name of countries parts when the cursor hovers.

4.1.5. MATRIX PLOT

The matrix plot is an extension of a bar chart where instead of visualizing one categorical and one numerical variable, we can visualize two categorical and one numerical variables. The matrix plot constructed is very analogous to a stacked bar chart however, instead of using height as a primary feature to encode information the area of a circle is used. Thus when a merge occurs a pie chart is formed consisting of the merged categories. A zoom can then be performed highlighting individual arcs of the pie chart. This was done to follow the data visualization mantra of overview, zoom, and filter then details on demand . Additionally, a tooltip is used to relay information to the user about the size of each individual slice present within the pie chart. With the incorporation of a matrix plot, the user is given the option to compare relations between various attributes. To be more specific within the domain of the Airbnb data-set, the user is able to make comparisons between the different types of rooms available in the different cities. Thus this extension allows the user to perform a different set of functions that were not possible while using the original system. This is illustrated in detail in the case study presented below.

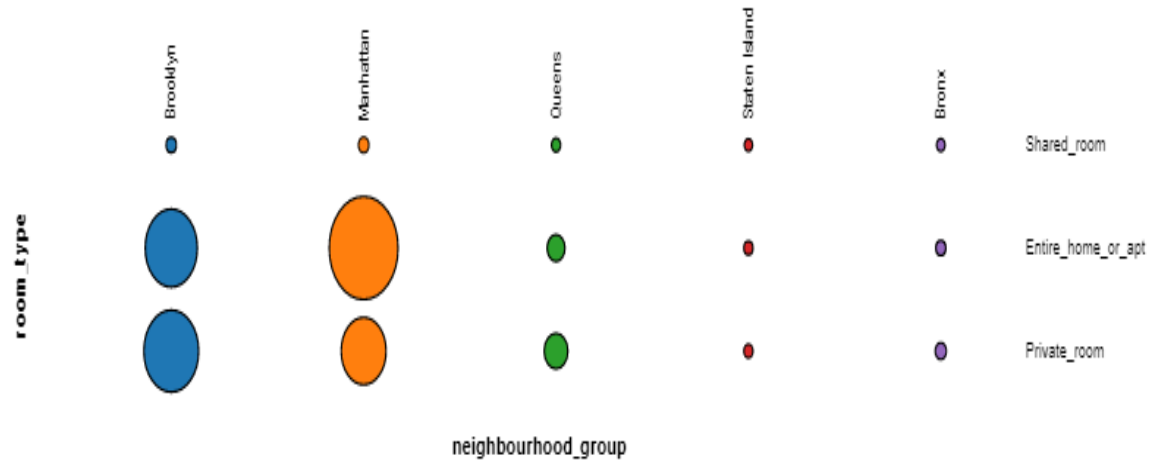

Figure 4.6 Matrix Plot

4.1.5.1.Steps in Matrix plot

Step 1: Start with creating the SVG.

Step 2: Let's load our data from the CSV file.

Step 3: Next, define which two categorical data and which numerical data is to be compared based on the confidence level.

Step 4: Set the dimensions and scale for each pie graph.

Step 5: Fill the charts and add a zoom operation for making visualization clear.

Step 6: Add a tooltip to relay the information about the size of each individual slice in the pie chart.

4.2 RESULT

4.2.1 House Price Prediction DataSet:

Figure 4.7- Output-1

House Price Prediction had details about different houses in India. Let us see how easy it is to get an overview of this dataset using our system. Let us first focus on the distributions. Using the bar charts and histograms we can easily get a general overview of all of the visualized attributes in the dataset. However, looking at the “Posted_By” and “BHK_OR_RK”, the analyst might wonder, “how are the

different types of rooms distributed across say, Dealer?". Looking at these visualizations we can't really answer this question.

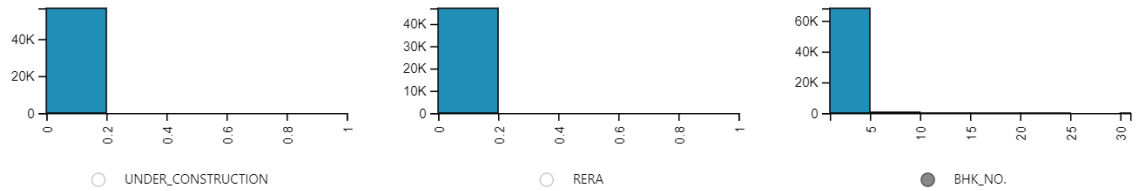

Figure 4.7.1- Bar Graphs

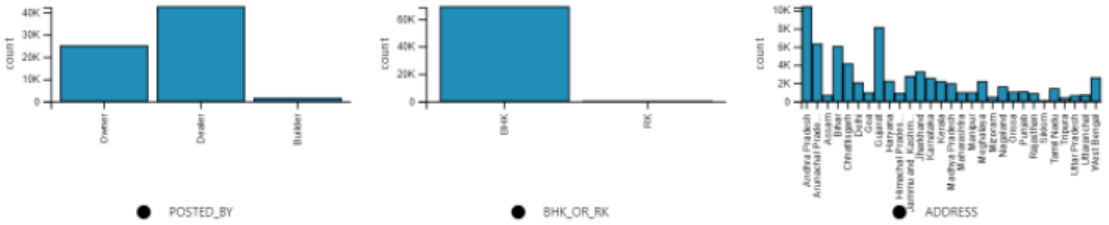

Figure 4.7.2- Histograms

The analyst can look at the Matrix plot. In this plot, the analyst has a pretty good idea that within Manhattan, maximum types of properties are of the type "Entire_room_or_apt". The analyst now looks at the distribution of Dealer and notices that Dealer has a really high number of BHK rooms. He might wonder if the number of RK rooms posted by dealer, owner and builder are the same, since they look pretty much the same size. This is where our extension comes into practice. To be sure of her hypothesis, the analyst performs a merge in the Matrix plot.

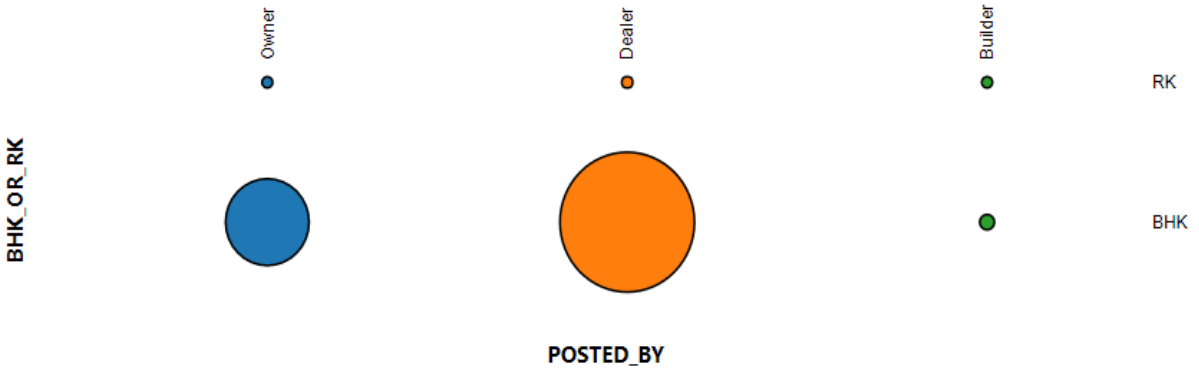

Figure 4.7.3- Matrix Plot

Looking at the updated plot, the analyst is now sure that the numbers are not actually the same. The number of RK rooms posted by builders is low as compared to the number of RK rooms posted by owners and dealers. The analyst can perform a number of similar comparisons between data across two different categorical attributes in the dataset.

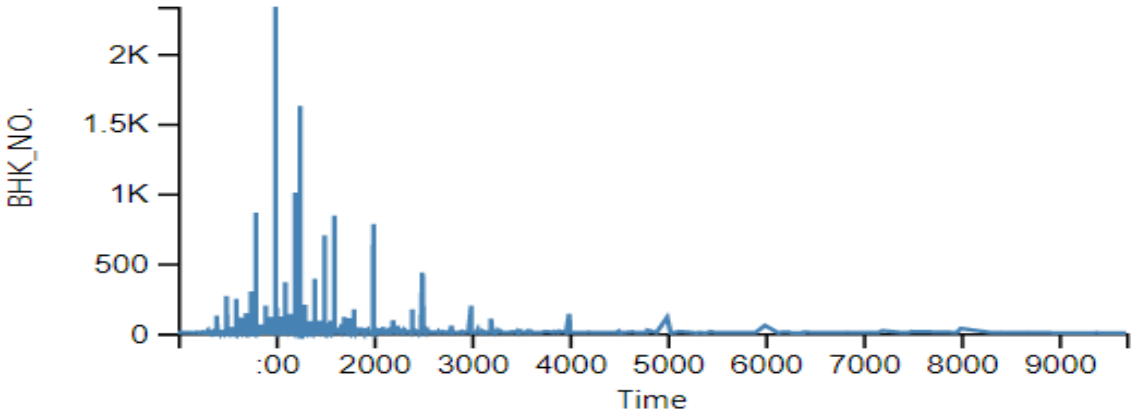

Figure 4.7.4- Line Graph

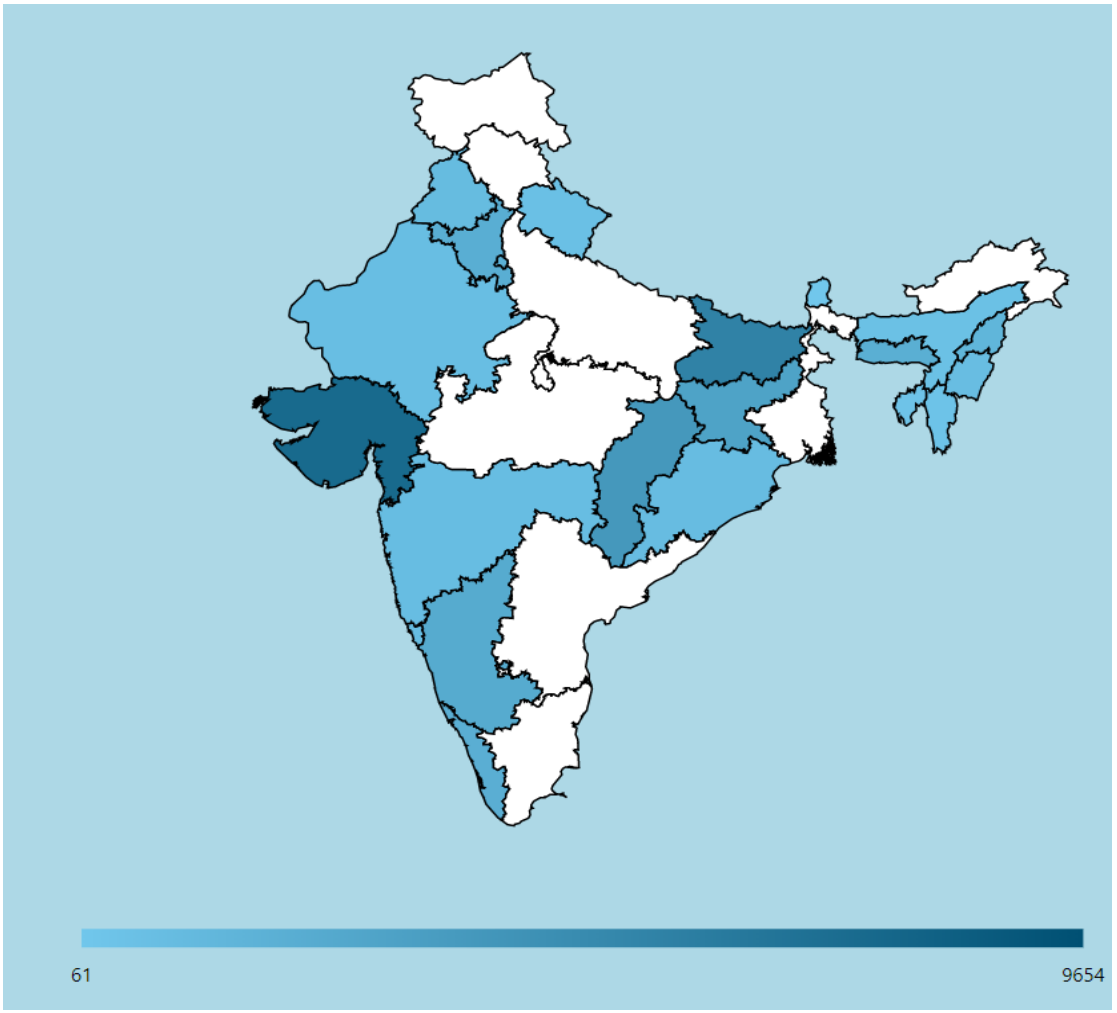

Figure 4.7.5- Map View

4.2.2 New York City Airbnb Open DataSet

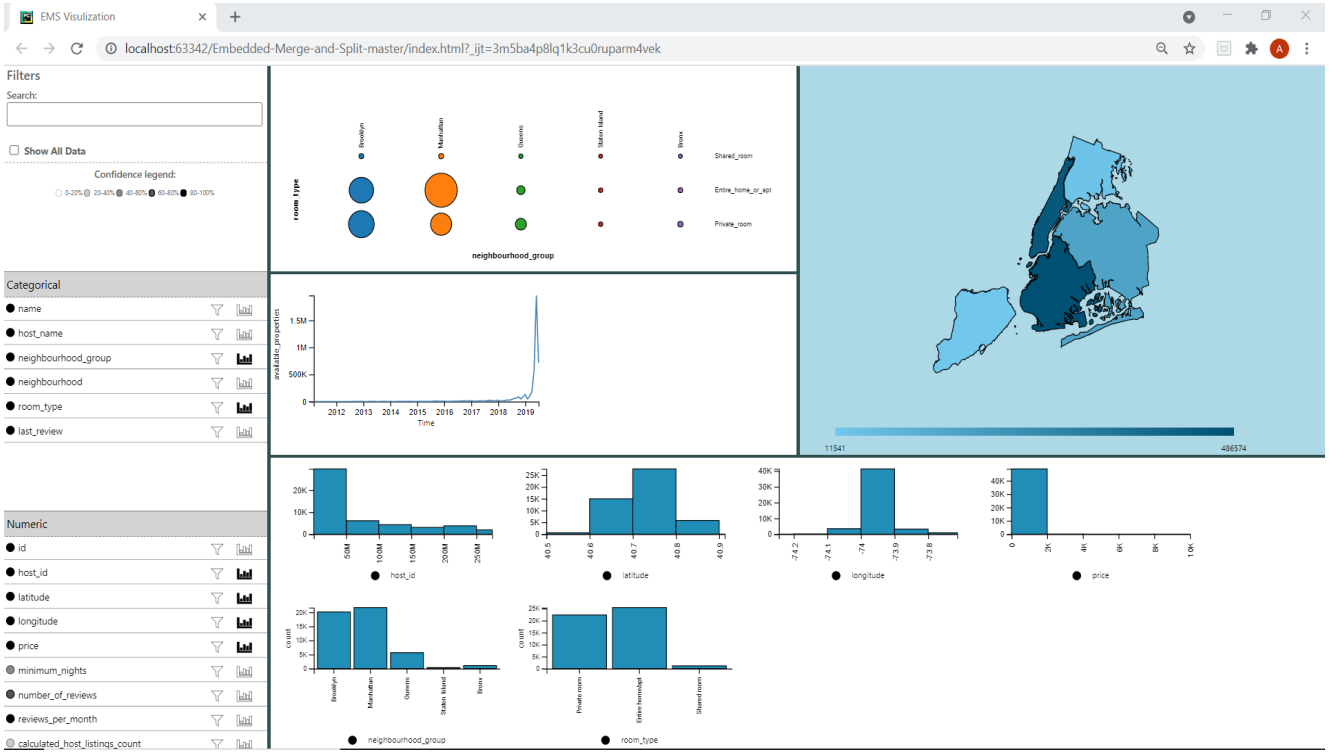

Figure 4.8 Output-2

Airbnb New York City had details about different properties in the New York City region. It shows how the system looks like when this dataset is loaded into it. Let us see how easy it is to get an overview of this dataset using our system. Let us first focus on the distributions. Using the bar charts and histograms we can easily get a general overview of all of the visualized attributes in the dataset. However, looking at the “neighbourhood_group” and “room_type”, the analyst might wonder, “how are the different types of room_types distributed across say, Brooklyn?”. Looking at these visualizations we can't really answer this question.

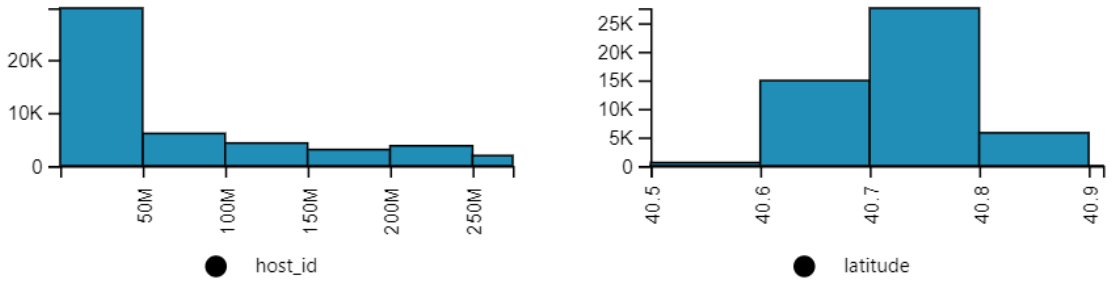

Figure 4.8.1- Histograms

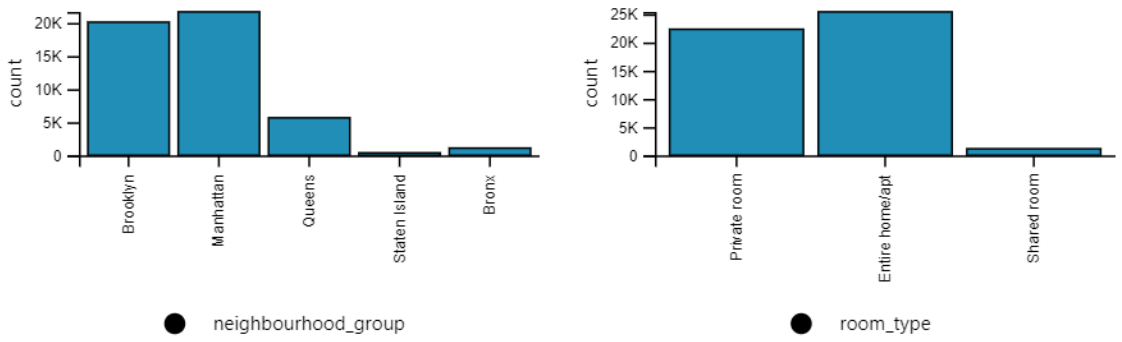

Figure 4.8.2- Bar Graphs

The analyst can look at the Matrix plot. In this plot, the analyst has a pretty good idea that within Manhattan, maximum types of properties are of the type "Entire_room_or_apt". The analyst now looks at the distribution of Brooklyn and notices that Brooklyn has a really high number of Private rooms. He might wonder, is the number of private rooms in Manhattan and Brooklyn the same, since they look pretty much the same size. This is where our extension comes into practice. To be sure of her hypothesis, the analyst performs a merge of Brooklyn and Manhattan in the Matrix plot.

Figure 4.8.3- Matrix Plot

Looking at the updated plot, the analyst is now sure that the numbers are not actually the same. The number of private rooms in Manhattan is low as compared to the number of private rooms in Manhattan. The analyst can perform a number of similar comparisons between data across two different categorical attributes in the dataset.

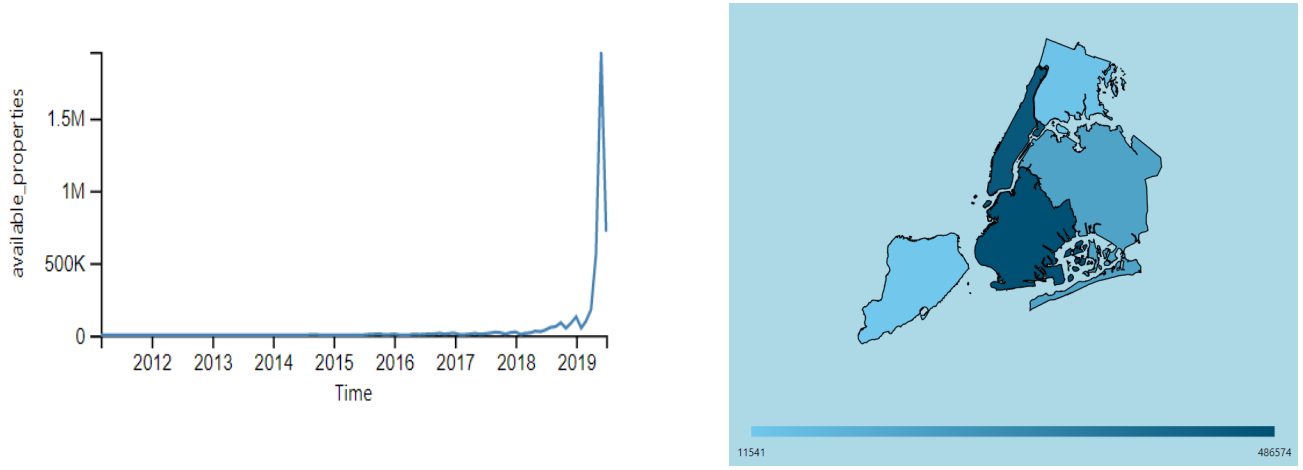

Figure 4.8.4- Map View & Line Graph

4.2.3 Border Crossing Entry DataSet:

Figure 4.9 Output-3

The US Border crossing entry dataset had all of the statistics about inbound crossings at the two borders of the US: USCanada border and US-Mexico border. Here as well the analyst might want to get a general overview of the dataset before generating any hypothesis of her own. Looking at the map plot, and applying relevant filtering operations in bar charts and histograms, the analyst initially came

to the conclusion that people entering from the US-Canada border entered through the northern states of the US, and people entering from the US-Canada border entered through the southern states of US. This actually made sense even without a visualization.

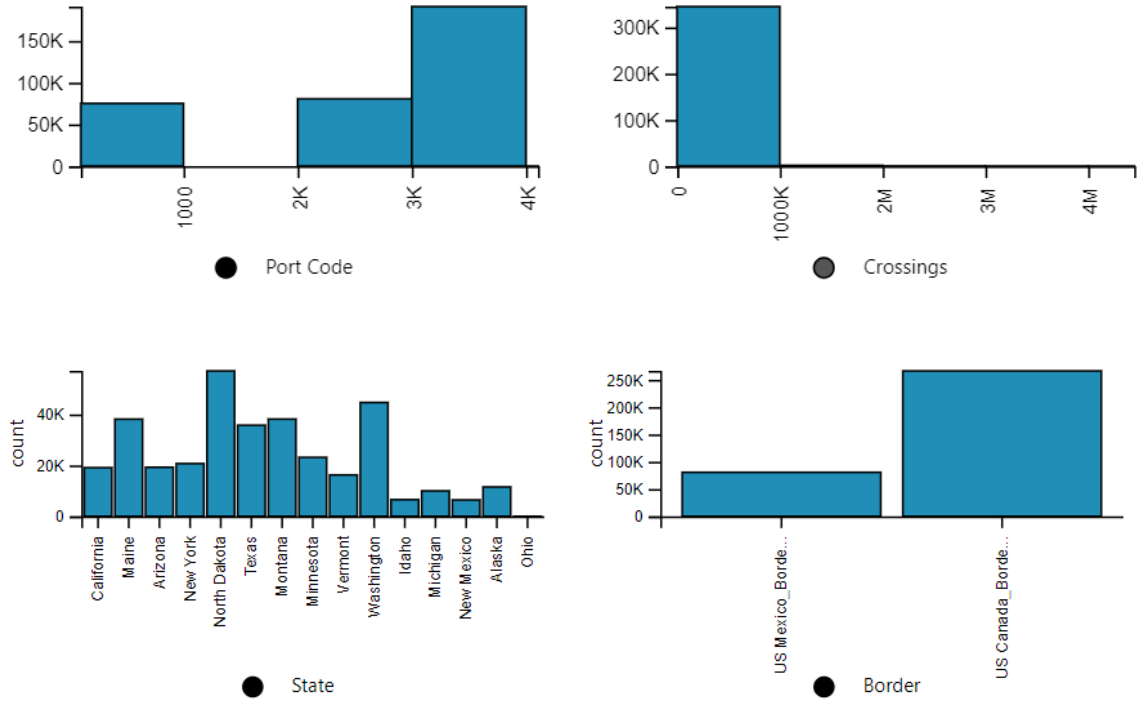

Figure 4.9.1- Bar Graphs & Histograms

Looking at this distribution, the analyst might wonder: "What is the most common way of transportation (Measure) when traveling to different states?". She, first of all, looks at the plot of Measure and sees that all of the modes of transportation used are almost uniformly distributed. She then looks at the plot of Border and sees that there were a lot more numbers of border-crossings from the US-Canada border than from the US-Mexico border. Therefore, she might hypothesize that people entering the southern states, since they came from Mexico with higher temperatures as compared to Canada, must have a higher number of transportation counts for heavy vehicles like trains, buses, etc.. To confirm her hypothesis, she again looks at the matrix plot, the values seem really similar, therefore the analyst combines the plot for Train with "Personal vehicle".

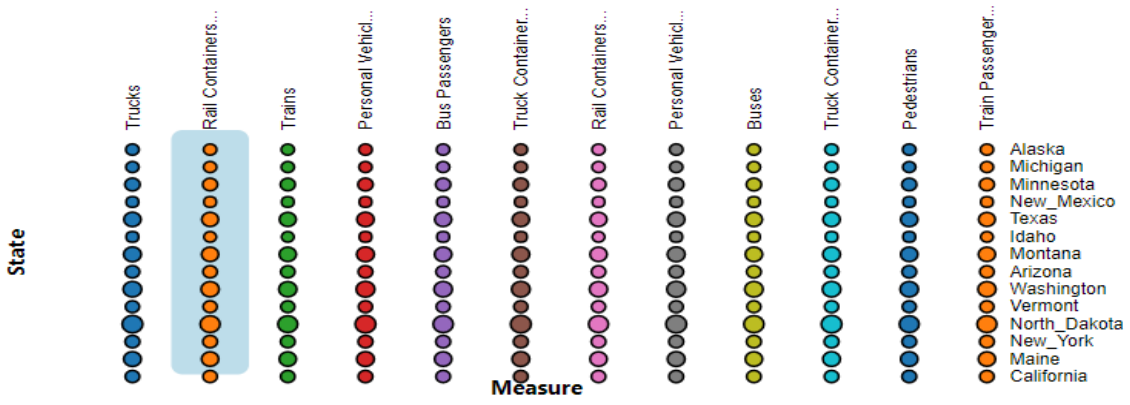

Figure 4.9.2- Matrix Plot

Looking at this plot, the analyst faces the issue of small data marks. She can't actually compare the values as easily as she did in the Airbnb dataset. However, our extension supports this issue as well. The analyst can hover over any of the pie-chart and it zooms-in. Looking at this zoomed-in view, the analyst can see that personal vehicles were used almost the same number of times as trains. She can similarly look at other modes of transportations as well to confirm or reject her hypothesis.

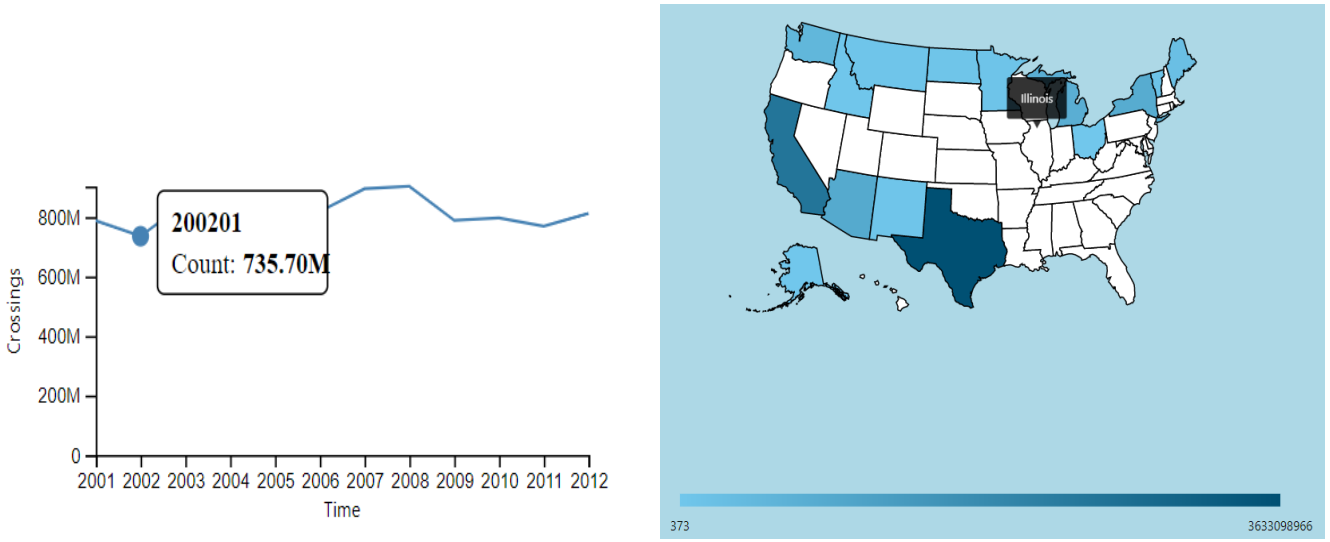

Figure 4.9.3- Line Graph & Map View

4.2.4 Suicides In India DataSet:

Figure 4.10 Output-4

Suicides in India dataset had all of the statistics about the suicides attempted in India between 2001-2012. Here as well the analyst might want to get a general overview of the dataset before generating any hypothesis of her own. Looking at the map plot, and applying relevant filtering operations in bar charts and histograms, the analyst initially came to the conclusion that people of uttaranchal are committing more suicide attempts. This actually made sense even without a visualization.

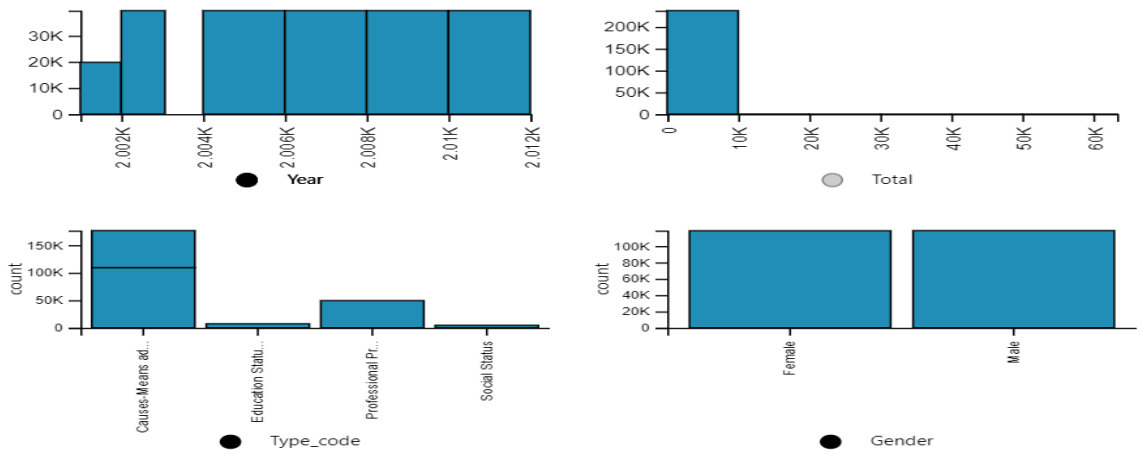

Figure 4.10.1- Bar Graphs & Histograms

Looking at this distribution, the analyst might wonder: “What is the most common reason of suicide(Type_code) in each gender?”. She, first of all, looks at the plot of Type_code and sees that all of the causes are high. She then looks at the plot of Gender and sees that they are uniformly distributed between male and female. Therefore, she might hypothesize that people from Uttaranchal have more cause issues and are committed by both genders equally. To confirm her hypothesis, she again looks at the matrix plot, the values seem really similar, therefore the analyst combines the plot for genders”.

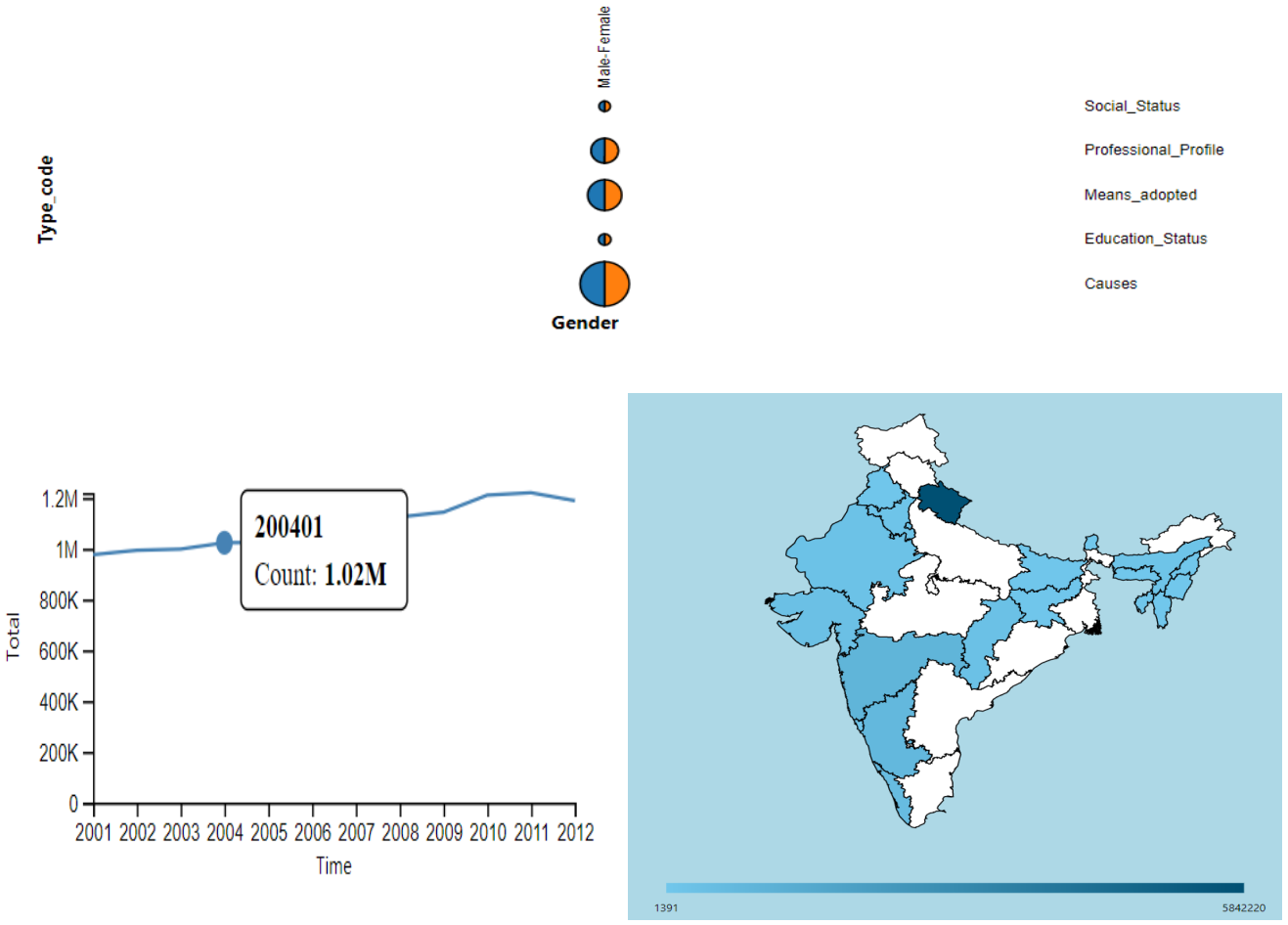

Figure 4.10.2. Matrix Plot , Line Graph & Map View

Looking at this plot, the analyst faces the issue of small data marks in social_status. She can’t actually compare the values as easily as she did in the causes. However, our extension supports this issue as well. The analyst can hover over any of the pie-chart and it zooms-in. Looking at this zoomed-in view, the analyst can see that both are equal. She can similarly look at other type_codes as well to confirm or reject her hypothesis.

5.CONCLUSION AND FUTURE SCOPE

5.1 Conclusion

We implemented and evaluated EMS for bar charts, histograms, maps, line graphs and matrix plots providing initial evidence that our technique can facilitate adjustment of data grouping. Embedded Merge & Split(EMS), a novel embedded interaction technique that enables users to adjust data grouping criteria by directly manipulating encodings used for presenting groups. Compared with tableau software EMS technique can significantly reduce the time required to merge, split and change the number of groups. It simplifies the complex data and identifies relationships more effectively. This work is the first step towards exploring the relatively large design space of data grouping in visualization tools.

5.2 Future Scope:

Multiple avenues for future work lie in enriching the EMS technique. We envision expanding the EMS technique to other visualizations and data types and work towards expanding EMS to other devices. Even our future work may include extensions that could be useful for user accessibility such as extending the system to support further visualizations that include bubble charts, pie-charts, etc. Interaction of “changing the number of bins” in a Bar chart can be implemented as going from a higher category to a lower category. This can be achieved using a simple mouse scroll (unlike the case in histograms) or merging two or more groups when reducing the number of bins. Currently EMS is not supported in touch-based devices. We can use a tool-tip that shows a zoomed-in view of the finger position on the screen.

REFERENCES

1. For installing visual studios:
   https://visualstudio.microsoft.com/download.
2. For D3:
   https://d3js.org/
3. https://www.kaggle.com/dgomonov/new-york-city-airbnb-opendata
4. https://www.kaggle.com/akhilv11/border-crossing-entrydata
5. https://www.cc.gatech.edu/~aendert3/resources/Sarvghad2018Embedded.pdf
6. https://www.kaggle.com/rajanand/suicides-in-india
7. https://www.kaggle.com/anmolkumar/house-price-prediction-challenge/tasks?taskId=2304
8. https://medium.com/multiple-views-visualization-researchexplained/what-is-visualization-research-what-should-it-be-8840a9ba658