KING'S College LONDON

University of London

Prototype for a population visualization tool

Final Project Report

Author: Jan Aldous Torres

Supervisor: Dr Simon Miles

Student ID: 1323454

April 17, 2017

Abstract

(Will be filled after the completion of the report). The abstract is a very brief summary of the report's contents. It should be about half-a-page long. Somebody unfamiliar with your project should have a good idea of what your work is about by reading the abstract alone.

Originality Avowal

I verify that I am the sole author of this report, except where explicitly stated to the contrary. I grant the right to King's College London to make paper and electronic copies of the submitted work for purposes of marking, plagiarism detection and archival, and to upload a copy of the work to Turnitin or another trusted plagiarism detection service. I confirm this report does not exceed 25,000 words.

Jan Aldous Torres

April 17, 2017

Acknowledgements

I would like to thank my supervisor Dr. Simon Miles for his support in this project. It is much appreciated.

I thank Lambeth Council for their support in the requirements and evaluation stages of the project. Taking time off their busy schedules was much appreciated.

Contents

List of Figures

Chapter 1

Introduction

[**?**] [5] [6] [7] [**?**] [9] [1] [2] [8] [4] [3]

Local government councils are challenged with the recent cuts in funding and need to cater to a diverse population with different needs. Therefore, there is a need to efficiently allocate resources. There has been a shift in local government, which is promoted by the national government, to utilize the concept of customer segmentation in marketing. This will let them understand the needs of segments of the population better thus letting them utilize and allocate resources in the most effective way based on those needs [4]. This approach would require the council to identify population groups with similar needs. In government, using data analysis in policy decision making is becoming more popular(new york guy video)

Clustering algorithms groups a data set into groups in which elements in a group are more similar than other groups. Clustering will provide a more complete perspective into the different situations groups face instead of viewing a group as one entity and defines variables which distinguish the group from the rest of the population using clustering algorithms. This is the first step in customer segmentation where the algorithm can identify groups which may not have been evident before. The visualization of the clustering results make data on groups are more accessible and legible to the user.

There are off-the-shelf software packages available which supports users in customer segmentation from clustering, in a statistical sense, to the visualization of their data. The Local Government Agency has created a guideline to support councils in tailor-made clustering and visualization [4]. Moreover, there have been projects by other local councils to visualizations of data on prominent population groups.

However, these tools require technical skills which the policy makers may not have or only

uses a specific data set. Those in councils who deal with policy making who deal with such problems also deal with numerous data sets. Analyzing these data sets is a tedious task that requires technical expertise and distances the policy makers from exploring the data themselves.

This project aims to produce software so that the commissioners could explore the data themselves through visualization of the data and using clustering algorithms used in customer segmentation. This project aims to implement a prototype of such a tool as a Django web application.

The tool allows the user to create population groups and visualize data about them. Visualizations of this data will also help to compare characteristics of the group to the wider population and subgroups. This comparison may lead to observations on the circumstances a group or subgroup is facing. This may lead to the conclusion that the council may want to increase or tailor resources allocated to a group or service. Since the user may not necessarily know what they are looking for when they view the data, the visualizations will be in the form that will let the user explore more aspects of the data. The tool aims to aid policy makers to explore data about their population and identify which problems segments of the population are facing. Thus it could help make better decisions on the allocation of their resources.

1.1 Report Structure

The remainder of this paper is organized as follows. Chapter 2 introduces the concept of customer segmentation, clustering, processes of data analysis and related work to the tool. Chapter 3 discusses the requirements of the tool. Chapter 4 outlines the specification and design of the tool. Chapter 5 describes how the tool was implemented. Chapter 6 discuses the professional and ethical issues. Chapter 7 describes the results of the evaluation after the tool was demonstrated to members of the policy department. Lastly, chapter 8 discuses the conclusions and future work of the project.

Chapter 2

Background

This chapter includes a background on customer segmentation and clustering algorithms, followed by a background on the use of data analysis, data analysis in government, data visualization and exploration. The chapter ends with a section on related applications to the tool.

2.1 Customer segmentation

Customer segmentation or market segmentation tool is widely used in the marketing field. The whole market is divided such that the each segment has similar "needs, wants or demand". The organization uses it to target a segment to tailor services to their needs.

The four basic market segmentation-strategies: behavioral, based on a person's behavior and decision-making process, demographic, based on the background of an individual such as age, gender, nationality, psychographic, similar to behavioral but is more about the lifestyle and interests, and geographical, based on a person's location. (market91) Local governments usually uses demographic segmentation as suggested in () more than the other types (). There are other more complex segmentation approaches involving the customer's interaction with a service or access to a service. However, they are not as prevalent in the public sector (citation). Clustering algorithms could also be used to identify the segments (lga).

The resources of an organization are limited and it usually cannot cater for the whole population [citation]. The private and public sector have to approach customer segmentation differently as the private sector aims to maximize business and targets the specific segments of the population. This is contrary to the public sector which has the mandate to cater to the whole population. They focus more on offering appropriate services to segments of the

population. Policies usually targets segments of the population, where the minorities may not experience the benefits and only the majority of the population do.

Customer segmentation is a concept promoted by the national government and the Local Government Association (LGA) has created a guidance document for any council that wishes to implement such a tool [4] [1]. They give guidance on which data sets to use, segmentation of the population with k-means clustering and ways to visualize the data. [1] suggested the use of explicit data about its customers and implicit data âĂIJknowledge of staff on customers using a serviceâĂİ. This would give a more complete view on not only the customersâĂŹ profile but also their behaviors in interacting with the service. .

2.2 Clustering

Clustering is a task which aims to group data points in an unlabeled data set where each point within a group, called a cluster, is more similar to another point in another cluster. It can be implemented as different algorithms using varying methods of determining similarity. The number of clusters is determined by the user depending on their expectations from the data(citation). It has numerous applications in different fields such as data science, biology, medicine and business to name a few. It is also a technique used in customer segmentation where a clustering algorithm can label the segments within a collection of market data (lga).

The optimal number of clusters is subjective however, there are ways to determine the best number of clusters to use. One method, the elbow method, requires a graph of the percentage of variance and the number of clusters (citation). The sum of the intra-cluster distances between points in a cluster. The normalized intra-cluster sum of squares gives the variance quantity. The percentage variance is calculated as the ratio of the between-group variance to the total variance (). As the number of clusters increases the percentage of variance increases significantly until a certain point, which creates a bend. In this method, the first bend in the graph determines most suitable number of clusters, however finding the bend may be an ambiguous task as it depends on the clarity of the bend.

There are two main types of clustering, partitional and hierarchical. Partitional clustering begins with one cluster and repetitively clusters points to the nearest point or cluster until it reaches the desired number of clusters. Hierarchical clustering begins with each data as its own cluster and repetitively merges the most similar clusters or points into one cluster until it reaches the desired number of clusters.

There are different measures to determine if data points should form a cluster which include

the structure of a cluster through closeness but also the concept of the cluster. Distance, metrics such as Euclidian, cosine, Jaccard, Hamming distance, Manhattan (citation)(doing data science). An implementation of this is the k-means algorithm which uses Euclidian distance to calculate similarity. However, it is not as preferable for clustering categorical data. Euclidian distance calculates a mean which could relate two discrete values or categories together. HuangâĂŹs k-modes (citation) is an algorithm to avoid this issue. It instead matches categorical data of different data points through the calculation of the mode, respecting the categories as discrete values, rather than the contrary in Euclidian distance.

There are other ways of clustering through structure which involve density, which includes a point such that it does not exceed density of points within the radius of a cluster. Conceptual clustering uses a descriptive language to define the concepts of the cluster through instead of similarity measures (citation). The descriptive language defines the model in which to compare the similarity of data points.

2.3 Data science

Data science is a multi-disciplinary field which intends to âĂIJanalyze and understand actual phenomenaâĂİ (springer) but it also deals with the communication of its results (doing data science). It is multi-disciplinary because it not only involves statistical data analysis, but also domain expertise, data visualization, machine learning which are skills needed in this field (data science).

In the field of data science, data exploration is one of the steps in the data science process. After data is collected from the real world, processed and cleaned, which converts raw data into a usable format. Depending on the process used, the next step is either data exploration (data science) or stating the question followed by data exploration (art of).

Nevertheless, though the order may be different, both process agree that data science involves exploring the data, building models, interpreting and then communicating the results. These steps demonstrate the need for different disciplines in this field as statistics and domain expertise is required in building models and data visualization is required for communicating the results.

2.3.1 Data exploration

Data exploration aids the user in understanding the structure of the data set, discovering if there are missing values, examining the distributions of individual variables, to name a few (art of). (data science) explains that data exploration is not only about confirming expectations or hypotheses but also to discover new information which one did not expect. Tools include plots, graphs and summary statistics. Data visualization through plots such as boxplots and graphs allows one to absorb information and identify patterns more easily. Summary statistics involve simple statistical data such as a variableâĂŹs mean, median, mode and range.

2.4 Related works

This subsection describes some works related to the software created in this project, which includes academic research of prototypes for data exploration, customer segmentation tools and government websites which visualize data. There are other related tools, however, reviewing all of them is beyond the scope of this report.

2.4.1 Data exploration tools

Gogolou et al. [9] created prototype, Data Curation and Validation (DCV) which supports user-data interaction especially in data cleaning and exploration. DCV, originally a data cleaning tool, has lead them to believe that any interaction with the data is data exploration. Therefore, the tool has been expanded to include data exploration features. There are three components, the presentation, visualization and profiling. The presentation component allows the user to select the type of visualization they wish to see on a variable. The profiling component predicts if there is erroneous data such as outliers and recommends actions to the user according to previous data analysis. The activity component deals with data exploration tasks such as data search and analysis.

Similarly, Graves and Hendler [10] created a prototype which allows users with no technical expertise, but more specifically targeted to visualize open government data (OGD). Ordinary citizens, journalists, those with no experience in consuming OGD are at a disadvantage. Visualizations enables an ordinary person to consume large amounts of data. Therefore, creating visualizations of OGD will enable them to consume this data. They identified groups of software which deal with this problem namely, office suits, business intelligence software, specialized analysis tools and visualization APIs which are not suitable for the consumption of OGD by

the layman. They discuss what is needed in such a software: simplification of the visualization process, providing meta-data about the current data set such as location, time of creation, understandable conventions on description of data (i.e. turn FY1998 into its worded description), including the contact information of the provider, sharing visualizations on social networks.

The neighbourhood statistics website of the UK government gives a summary and visualizes some census data. The website enables the user to view census data on a geographical area. The user can view a data set as a table or on a map (see figure). The table shows aggregation of the data set where each row represents a variable. The map visualizes the values of the selected variable through different shades of color according to the legend. A time series graph also allows the user to view differences of the aggregate value over the years. The tool however limits the user to viewing only one variable at a time.

Kent and Medway has tool which segments its population by social class and aims to highlight the key features which make each Group distinctive, to help you visualise the segmentation data and understand the essence of each Group (Kent Medway). It lets the user compare the values between all groups. This tool is bespoke to Kent MedwayâĂŹs data and the tool cannot be reused for other councilâĂŹs data.

2.4.2 Customer segmentation tools

There are tools which presents premade customer segmentation analysis of the UK population like Mosaic Public Sector by Experian [3], Acorn by CACI [6] and Kent & MedwayâĂŹs interactive guide.

Mosaic Public Sector uses government data and visualizes this data through an interactive tool intended for public sector use. The tool has pre-made customer groups using their data and the visualizations include maps, photos and graphs. It utilizes data from different sources, from census to social media data to create the segments.

On the other hand, Acorn uses government but also commercially available data. Like Mosaic, it has created premade customer segments. However, there is a feature that enables the user to find out more about their customer through just their post code. This approach separates itself from matching demographic classifications to locations to matching locations to demographic types.

Kent & Medway tool which visualizes the segments its population by social class determined by their own customer segmentation analysis. It has a feature which compares a data variable between all groups of the population (see figure)

There have been similarities between implemented tools and recommendations to differentiate and describe each group. [5] [4] suggested to include maps showing the concentration of that group in each ward and in addition to a textual description, they include pictures to describe each group. [2] has a report on the customer segmentation of its population, and includes the percentage and textual description of its population. [5] includes graphs to visualize data but also word maps of key characteristics. [4] has suggested that the use of spider diagrams which detail the variables compared to the town and district average. Similarly [6] [5], they graph pieces of data with an index of the group compared to the overall population. This visualizes both the average value and the groups relation to that average displays whether they are higher or lower than that average however each graph is created for each variable.

Chapter 3

Requirements

The functional and non-functional requirements was created based on communication, through interviews and email, with a member of the Policy and Communications Team, and requirements which I have created based on initial requirements by our contact. Our contact described the context in which the tool may be used, the overall objective of the software, the users' technical abilities and some functional requirements, namely the method to create groups (see section 3.4). Communication with our contact was not sustained after initial requirements were collected due to the council's large workload. The rest of the requirements was created in accordance with the overall goal, initial requirements and according to my own analysis of the data.

3.1 Objective

The purpose of this application is to support those in local government councils, in charge of creating policies, to explore data on residents demographics and satisfaction with council services. User-defined groups will allow the user to focus on specific portions of the population. Aided by visualizations through graphs and maps, it may lead the user to identify which segments of the population requires the which resources.

3.2 Description of data to be used

The following local data provided by Lambeth should be used:

- **CSV of Lambeth's 2016 Residential Survey**: a survey of a sample of Lambeth's population consisting of 1024 people about their quality of life, what they thought of

Lambeth's services and about the respondents themselves. The data is in the form of a CSV text file. Residents'answers are categorical data in the form of code specified by the Survey Code Translation (see point below, Survey Code Translation). Single answer questions have its own column and its choice code is a positive integer. Multiple answer questions, each answer is regarded as a sub question in the form of 'Q5A' meaning question 5, choice code 'A' (see below Survey Code Translation). Therefore, a sub question has its own column and entries are in code as a 1 or 0 meaning yes or no respectively. There are also other fields which have been added to the original survey results such as group, subgroup, quintile, which is a result of previous data analysis.

- **Microsoft Word document of Lambeth's 2016 Residential Survey Code Translation**: a Microsoft Word document of the original survey with the original questions and code used in the data associated with the question. Under each question is the choice code and English meaning of the choice. The choice code is either a number for single answer questions (e.g. 1. Male, 2. Female) or letters for multiple answer questions (e.g. A. Access to nature, B. Activities for teenagers, etc.).
- **GeoJSON of Lambeth's Ward's Boundary Specifications**: downloaded from Lambeth's website, it is a GeoJSON, simple geographical features encoded as a JSON, of the geographical boundaries of each ward.

3.3 Functional Requirements

Functional requirements with the words textbfshould or textbfmust is a required feature. Requirements with the word textbfmay, is a desirable feature which may or may not be implemented should there not be enough development time.

1. Data storage requirements
   (a) 1. The system must save the actual files of the datasets (CSV, text files or JSON) and not access the file through a URL of another website.
2. Grouping requirements
   (a) The user should be able to create groups based on the parameters to the 5 factors (see section 3.4). A group's information should be kept in a database.
   (b) The user should be able to view data about a group through graphs.

i. Single answer questions or questions which requires only one answer should be visualized as stacked bar charts or pie charts.

ii. Multi-code questions or questions which requires more than one answer should be visualized as bar charts.

iii. Clicking data on a graph should display which wards have answered the selected question and choice on the map (see functional requirement 4).

iv. The questions under section Questions to be visualized should be visualized following requirements 3a and 3b.

(c) Like a population density map, which shows higher density concentrations of people as a darker color and lower density concentrations as a lighter color on particular areas of the map, the map should show the selected data (see functional requirement 3c). The map should be divided into the council's wards; the boundaries of each ward should be obvious. The color of the ward should depend on the number of residents who have the selected question and answer (see functional requirement 3c).

(d) The user should be able to compare values of a data variable between all groups such that groups should be compared to the average value of the variable in the whole population (i.e. compare the percentage of disabled people who are male to the percentage of the whole population who are male).

(e) 1. The main data used in the visualizations should be the CSV of Lambeth's Residential Survey and its Code Translation.

3. Clustering requirements

(a) The system should segment a group using a clustering algorithm.

(b) The system should display any statistical information (e.g. the mean of the answers of a question) on the differences between the clusters.

(c) The user should be able to compare information between the group's data and each of the clusters'data.

3.4 Creating groups

A group should be based on residents'answers to 5 questions in the residential survey. At least 1 question is required to form a group, therefore the rest of the 4 questions could have any

answer. The group will be composed of residents who match the required answer(s) to the required questions.

There are 5 questions in which a user can create a group:

1. Whether they have a disability or long term illness (Q43)
2. What type of benefits do they acquire (Q45A-Q45K)
3. Educational/employment activity (Q46)
4. Whether they are on the London Living Wage (Q47)
5. Housing tenure (i.e. council tenant, private owner, etc.) (Q35)

*Question number in the CSV residential survey is in brackets.

3.5 Questions to visualize

Once a group has been created, the user must be able to see the answers to the following survey questions:

1. What matters most to them most (Q5) (Multiple, top 3)
2. How was their last contact with the Council was made (Q26A- Q26G) (Multiple)
3. How they use the website (Q29) (Multiple)
4. What services they have used (Q39) (Multiple)
5. How they access the internet (Q50) (Multiple)
6. How well the changes have benefited them (Q11) (Single)
7. Whether they feel they belong to their neighborhood (Q13_R1) (Single)
8. Whether they value the friendships in their neighborhood (Q13_R2) (Single)
9. Whether they could approach a neighbor for advice (Q13_R3) (Single)
10. Whether neighbors help out each other (Q13_R4) (Single)
11. Whether they would be willing to work with others to improve their neighborhood (Q13_R5) (Single)
12. Whether they would join community events in their area (Q13_R6) (Single)

13. Whether they regularly stop and talk to people in their neighborhood (Q13_R7) (Single)

14. Whether they would speak highly of their neighborhood when asked (Q13_R8) (Single)

15. Gender (QGEN) (Single)

16. Age (QAGE) (Single)

17. Ethnicity (QETH) (Single)

*Question number in the residential survey is in brackets followed by whether they are a single answer question (Single) or multiple answer question (Multiple).

The answers to the questions above should be visualized using graphs for each group and their clusters based on functional requirement 3.

3.6 Non-functional requirements

Usability: The users for the program are the policy makers of the council, those responsible for formulating strategies to allocate resources for a council based on the presentation of data analysis given to them. They do not necessarily have technical skills, in terms of being able to operate applications, or advanced statistical analysis skills. Since the users are not technical savvy, the ease of use is imperative to the design of the user interface. The visualizations should also be easily interpreted.

In terms of maintainability, this version of the system is only a prototype to show the potentials of such a system, therefore there will not be a need for maintainability of the code. In terms of security, the data being used is anonymous therefore there will be no need for security measures for the data.

Chapter 4

Specification & Design

This section describes the design of the system as a website.

4.1 Platform

A web application as a platform to create visualizations takes advantage of the numerous open source visualization libraries and HTML, CSS and JavaScript are well equipped in creating interactive and visually pleasing interfaces. For these reasons, the application will take the form of a website.

More specifically the design will be implemented as a Django (citation) web application. Django is a popular Python web application. It is well documented and has numerous open source libraries which supports the creation of visualizations. Using this framework, third-party libraries could be utilized to produce JavaScript needed to create visualizations. Python's machine learning libraries will be used to manipulate and analyze data.

4.2 Preprocessing of data

Prior to the creation of the application, some preprocessing of the data must be done. The Microsoft Word document of the survey code translation in its current form is not able to be queried since it is just a Word document. It should be created as a formatted text file that is still readable for the user so that the system can query it and and in the future, the user can create the text file themselves. Inputting the translations in one file will be more user-friendly than inputting translations of individual questions through a normal form.

For this system design, since the user has only given the survey code translation as a Word document. I will create the formatted text file myself.

<Source information>
$
<Survey question code 1>|<Shortened question in English 1>|<Original question in English 1>
<SINGLE or MULTIPLE>
<Choice code>;<Choice in English>
...
<Choice code>;<Choice in English>
$
<Survey question code 2>|<Shortened question in English 2>|<Original question in English 2>
...

Figure 4.1: Formatted text file of survey code translation (See Appendix for actual text)

As seen above the text file is still readable as each question's data is separated by a '$' sign and each code is separated by a ';' or '—'. The question text is distinguishable from the choices text.

4.3 System architecture

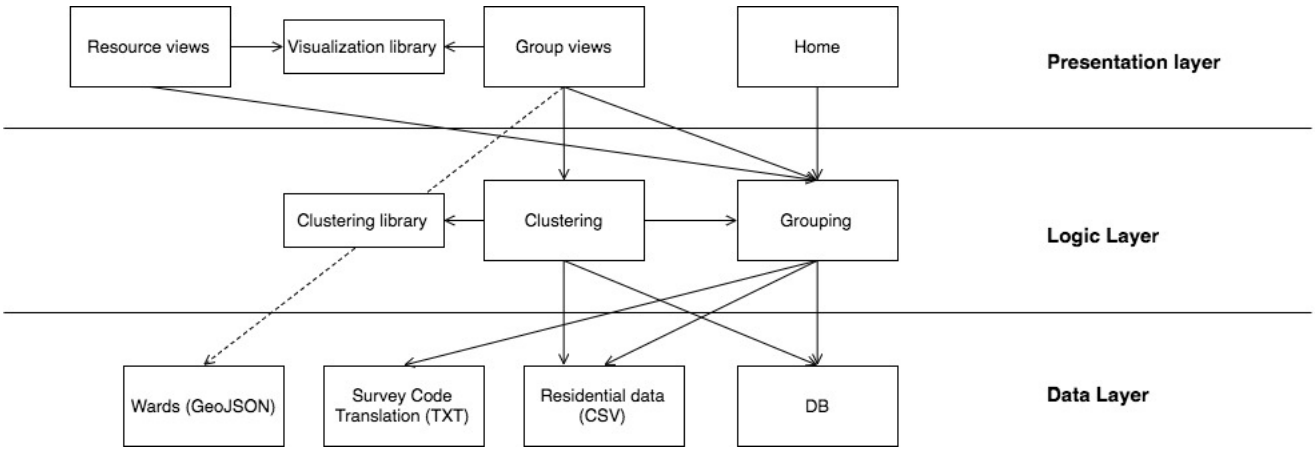

Figure 4.2: System architecture diagram

The system will follow the three-layer architecture.

The data layer will consist of the following components:

- **Ward**: a GeoJSON file containing the geographical boundaries of each ward
- **Survey Code Translation**: a text file formatted according to section 4.2 containing the translations of the survey code to English text

- **Residential data**: a CSV file of the 2016 Lambeth Residential Survey
- **DB**: the database which includes Groups created by the user and groups' Clusters produced by the clustering algorithms (see Appendix for more details)

The logic layer will consist of the following components:

- **Clustering library**: a third-party library used by the Clustering component to cluster group data
- **Clustering**: performs clustering on a group and stores results in the database
- **Grouping**: creates groups and extracts groups from survey data

The presentation layer will consist of the following components (see section 4.5 for more details):

- **Resource views**: shows visualizations data on each group for each question in the residential data set
- **Visualization library**: a third-party library used to create graph and map visualizations
- **Group views**: shows visualizations of groups and compares groups
- **Home**: main menu to choose which group to view

Though the original source of the Wards GeoJSON is from Lambeth's open data website, it will be downloaded as a JSON file and stored as part of the source code of the application. Since the ward boundaries is unlikely to changing, storing the file in the source code is preferable as it will not depend on another server's performance.

The Clustering component could be implemented separately and another clustering algorithm can be implemented without changing the Grouping component.

4.4 Algorithms

Data analysis will be carried out by the system through data clustering. The clusters will be generated by the Clustering component and will be saved to the database. The default number of clusters is 1 and the user will be able to manipulate the number of clusters.

There are numerous clustering algorithms however, due to the categorical nature of the data, k-modes clustering will be used instead of the the usual k-means clustering. K-modes has been proven to be more effective in clustering categorical data (citation).

4.5 User interface

The system's main goal is to present data and let the user interact with the user in an effective way. The following figures describe the interaction and design of the user interfaces.

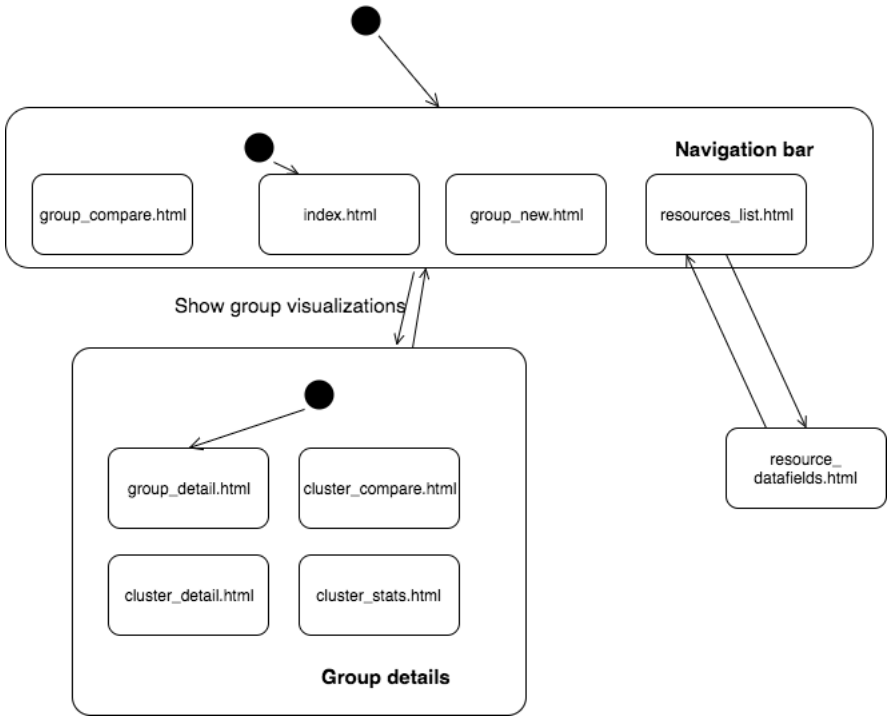

Figure 4.3: Interaction diagram of the web application

The interaction diagram above describes the flow between the pages in the website. It is designed so that switching pages is done in the least number of clicks possible. This will be done through a navigation bar where most pages will be accessible. The group detail pages will only be accessible from the group_detail.html page to avoid confusion with other groups' pages.

The following describes the functions of each page:

- **index.html**: contains a list of links to all groups
- **group_detail.html**: has visualizations of a group's data through maps and charts (see section for more detail)
- **group_compare.html**: compares all the groups to the average of the whole population of a data variable. This will visualize the comparison of the values of the cluster to the average value for that variable in the whole population (see section for more detail)
- **cluster_compare.html**: compares the data on the clusters of a group, to the group and the whole population for each question (see section for more detail)

- **cluster_stats.html**: lets the user change the number of clusters and shows differences, namely the means of an answer to a question, between each cluster (see section for more detail)
- **group_new.html**: contains a form that adds a new group to the database
- **resources_list.html**: lists links to residential survey questions
- **resources_datafields.html**: shows visualizations of data on all groups for a survey question

*Note that variable is to denote the selected question and choice (i.e. Q11, Full-time employee)

This design gives the user different perspectives on a population. Namely, a group's data (group_detail.html), data of clusters within a group (cluster_detail.html), differences between a group and its clusters (cluster_compare.html, cluster_stats.html) and differences between groups (resources_datafields.html). These features are intended to enable the user to identify interesting trends within a specific group, groups or cluster which may not be possible with fewer interfaces.

4.6 Interface design

The pages in the interaction diagram are described in more detail in the following page designs. Since a large part of the system is about the presentation of data, the layout of the visualizations and interactions within pages, shown as annotations, were designed.

The page on figure 4.4 is intended for the user to view the residential data which is visualized in graphs on the right-hand side of the page. Selecting a variable (i.e. a survey question and choice) on a graph will show its data on the map, on the left-hand side of the page, as a population density map. Each ward will be colored depending on the number of residents for which the resident chose the variable. This shows the user where residents who have chosen the variable live. Supporting the map is a bar chart of the number of residents for each ward who have chosen the variable which is ordered from greatest to least. This lets the user easily identify the distribution of the selected variable throughout the wards. The charts will follow functional requirements 3, which specifies which type of graph (e.g. bar chart, pie chart, etc.) to use for which type of question.

In addition, the meta-data about the data set used in the page is shown in the top left panel, which shows the number of residents in the group as a raw number and percentage and

Figure 4.4: Page design for group_detail.html and cluster_compare.html

the source of the data.

The same page design is used for cluster_compare.html except that the Question graphs will be in the form of stacked bar charts which show the question data on the whole population, the group and the clusters (see figure 4.5).

The figure 4.5 is the visualization for the resources_datafields.html page. However, the y-axis will instead be the groups in the database.

The page on figure 4.6 will let the user compare the percent difference from the whole population using the following equation:

(Percent of the group's population who answered selected question with selected choice-Percent of the overall population who answered selected question with selected choice)*100

Therefore groups more than 0% has a bigger proportion of their population which have answered the selected variable and vice versa for groups less than 0%. This may be an interesting

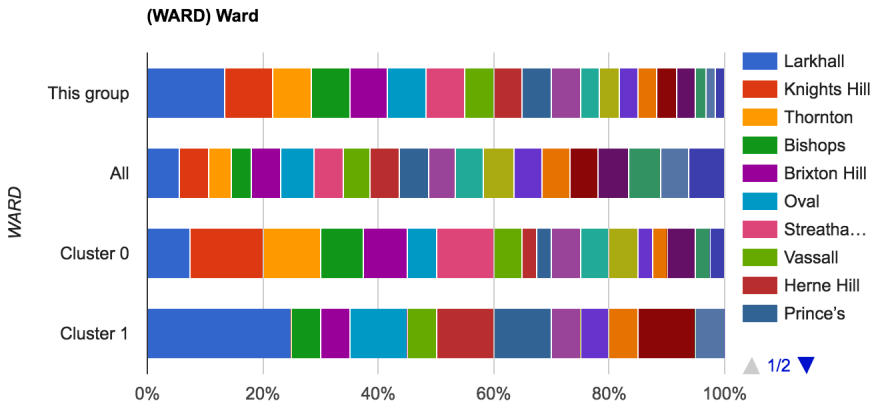

Figure 4.5: Chart for comparison of the data of the whole population, a group and its clusters

visualization for some questions more than others.

The page on figure 4.7 gives another perspective to the comparisons show in cluster_compare.html. Instead of showing the all the data variables, the set of answers are consolidated into a mean. The user can compare means of a question for each cluster which will highlight the differences between each cluster which is not always obvious in cluster_compare.html. The user may go back to cluster_compare.html to the data in more detail.

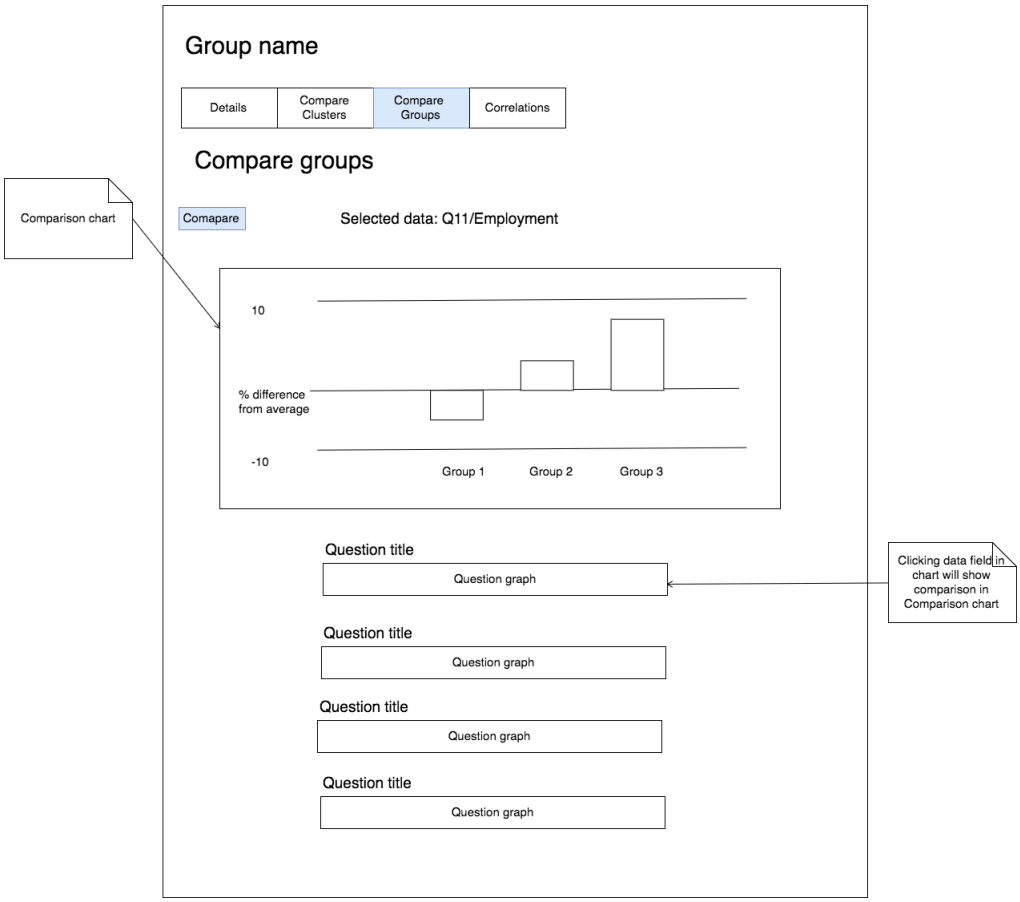

Figure 4.6: Page design for group_compare.html

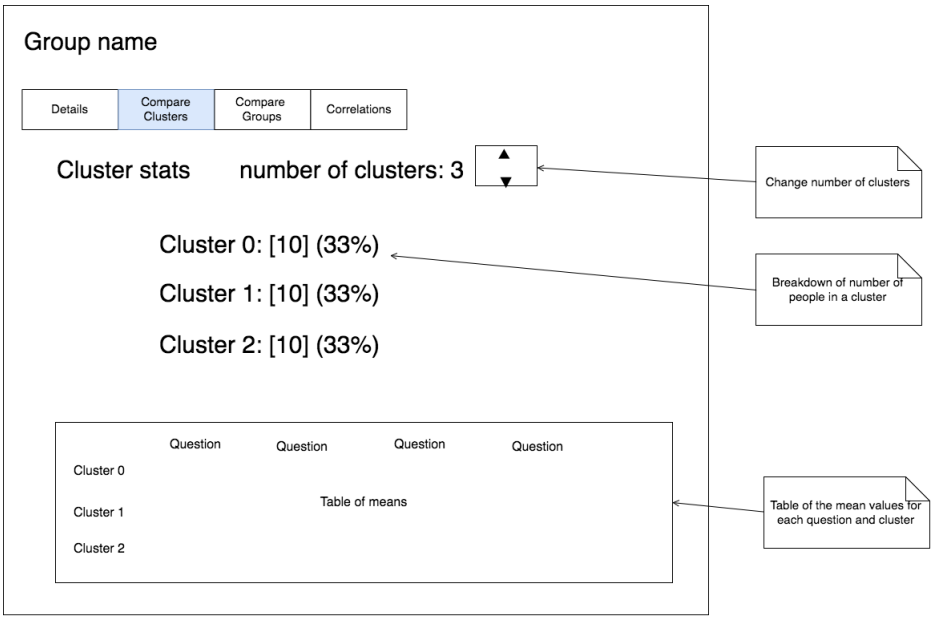

Figure 4.7: Page design for cluster_stats.html

Chapter 5

Implementation and Testing

This chapter explains the development approach used followed by how each element of the design was implemented.

5.1 Development approach

The software was developed through evolutionary prototyping. Since the initial requirements were not complete, this approach gave flexibility on what is to be implemented. Unlike throw-away prototyping where code is thrown away, this method begins with the requirements that are understood and seeks to incrementally implement additional features proposed by the user (citation). Though the usual definition is to modify the software based on feedback from the user, I have taken the place of the user and have incrementally designed and then implemented what is most likely suitable for the users' analysis of the data. Therefore, the interaction and interface designs of the application were created incrementally and the design as a whole was added in the Design chapter.

The following list outlines the iterations of the prototype including the requirements that were implemented:

- Implementation of groups in the population and graph visualizations of certain questions, group_detail.html and group_new.html using the residential survey data only (Functional Requirements 1, 2a, 2bi, 2bii, 2biv, 2e)
- Addition of survey code translations to the visualizations implemented in the iteration 1.
- Implementation of resources_list and resource_datafields.html

- Implementation of the map in group_detail.html (Functional Requirements 2iii, 2c)
- Implementation of the group_compare.html (Functional Requirement 2d)
- Implementation of clustering and cluster_detail.html (Functional Requirement 3a)
- Implementation of the interface cluster_stats (Functional Requirement 3b, 3c)
- Implementation of the interface cluster_compare.html (Funcitonal Requirement 3c)

The implementation stated with the initial requirements and each succeeding increment involved design, implementation and testing. The functional requirements were relatively done in order it was listed since each requirement usually depended on the previous requirement. The function and interface of each page was done together so that each prototype is functional. The prototype could then be tested and features for the next prototype could be suggested, which follows the spirit of evolutionary prototyping.

Development involved developing sections of the prototype that was independent of the system for some iterations. This was to familiarize myself to the different languages and libraries used in this application, which lead to the creation of an independently functioning section of the prototype. This was then integrated into the system to avoid confusion with other parts of the code. Independently developing it ensured that it was functional before it was integrated. This also reduced the risk of it not working. An example is the integration of a Google Map and the Google Charts used in group_detail.html. The clicking of a chart and display on the map was implemented in a separate HTML page and ensured that it worked. This was then integrated into the actual page, group_detail.html. The user friendliness was also improved through this method, as testing the interfaces in reality makes missing user friendly elements more obvious than could be seen in the design of the interfaces.

5.2 Third party libraries

The third party libraries are listed below:

- Django: a Python web framework used to create web applications
- Bootstrap: a CSS library used to make webpages more aesthetically pleasing
- Pandas (citation): a Python library used for data analysis and data structures
- Google Charts: a JavaScript API which creates different types of charts

- Google Maps: a JavaScript API which creates a map
- Graphos (citation): a Django app which turns data in the form of Python's list data structure into Google Charts JavaScript code âĂć K-modes: a Python library which applies a clustering algorithm most suited for categorical data on a set of data

Since it is a web application, a combination of programming languages were used, Python for the back-end with Django, HTML and CSS for the layout aesthetics of the web page and JavaScript to make the pages interactive.

5.3 Page implementations

The subsections below describe how each part of the software was implemented and some screenshots of the actual implementation.

5.3.1 Visualization of data into charts

Pandas Dataframe was used to store the Residential Survey data. The survey data as a CSV was loaded into a Dataframe and the latter was used to query the data around the system. A single instance of the Dataframe was kept to refrain from repetitively opening and reading the same file through the singleton design pattern.

Graphos was used to create the Google Charts. The required data of a chart was queried and formatted to Graphos' specficiations and a Graphos object was created. Since Django is a web framework, it has a template feature where Python objects, such as a Graphos graph object, could be turned into content for an HTML file.

Depending on the type of question, the charts on figure 5.1 was created.

As a default feature of Google Charts, hovering over a piece of data will display the value of data (see top left of figure 5.1).

5.3.2 Integration of survey code

Following the design for the preprocessing of data (see figure 5.4), the English Code translation text file was created. To speed up the creation of this file, the tables were copied into the text file and a small program was created to format the tables, whose elements are spaced by '\r', and replaced with a '\n' to format the spaces correctly. The rest of the file was typed manually.

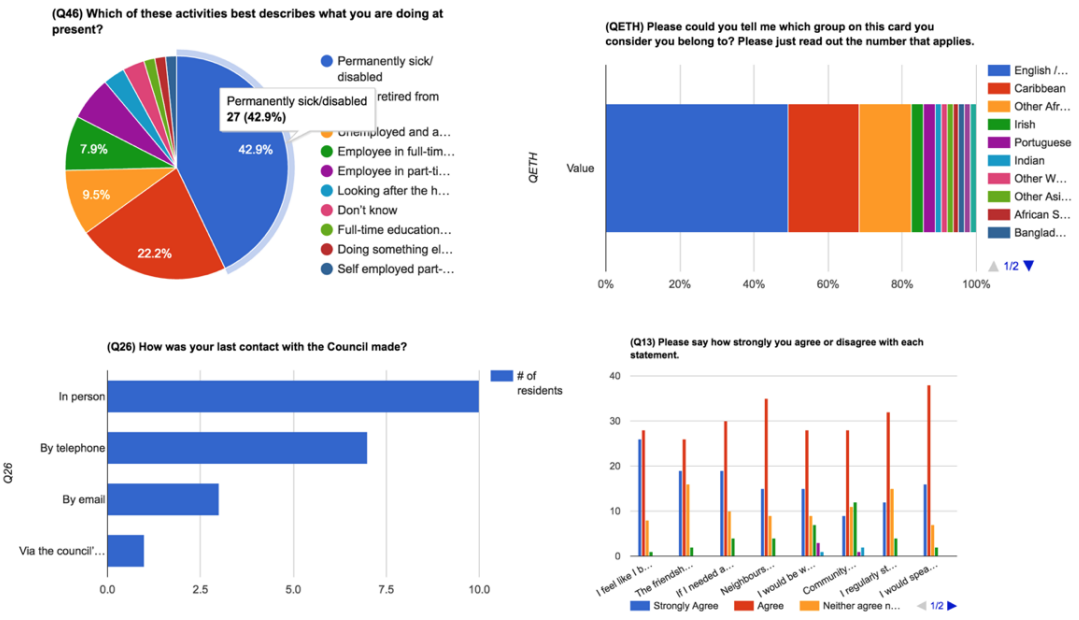

Figure 5.1: Implementation of the charts

Readtext.py is the Python module which reads and creates each question into Question objects. Readcsv.py consolidates English and code data into a Python list. This is inputted into a Graphos class which produces the Google Charts JavaScript code.

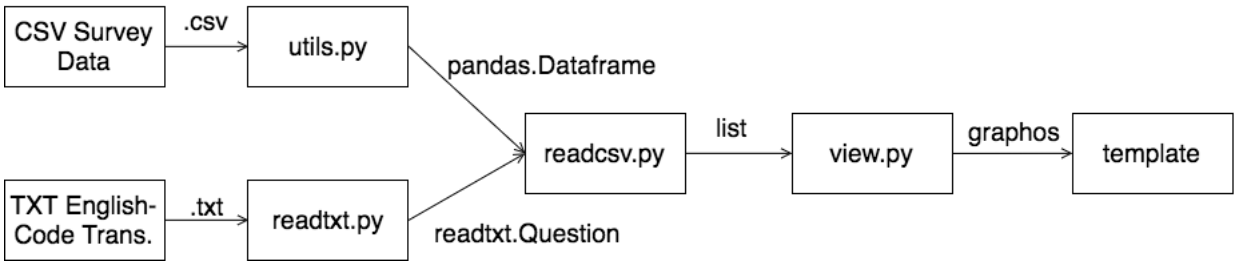

Figure 5.2: Implementation of the charts

Figure 5.4 shows the data flow from raw data to the creation of Graphos charts. This is a simplified view which omits other Django modules.

5.3.3 Visualization of wards on a map and ward breakdown chart

The map was adapted from a tutorial in the Google Maps website which implemented the display of colored areas depending on the value of that area. In this case, the areas were changed to Lambeth's wards.

The interactivity between the Google Map and Google Chart in the group_detail page was

developed as described in Development approach (see section 5.1).

Additional features to the Graphos code was added. A listener to see if the chart was clicked was added to Graphos' code. When a chart is clicked, the function which redraws the chart, map and changes the text is invoked.

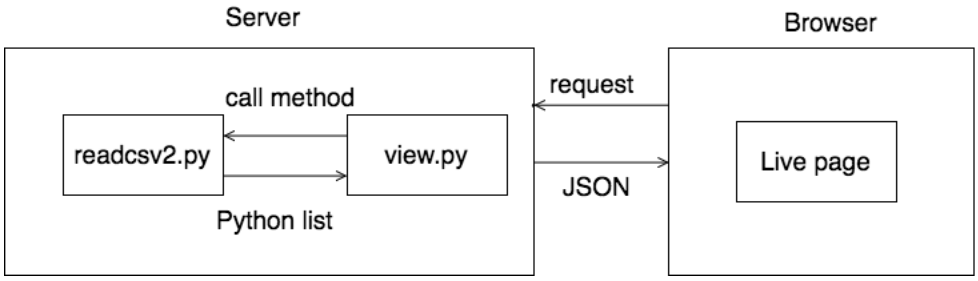

Figure 5.3: Implementation of the charts

This interactivity required the map's contents to be changed (i.e. change the colors of the wards depending on the question selected). A JSON containing a list, which is required in the creation of Google Maps, is populated with the selected data from a chart (see figure **??**). This JSON is requested by clicking a chart on the right hand side of the page, which invokes the redrawing of the map. The figure below explains the data flow.

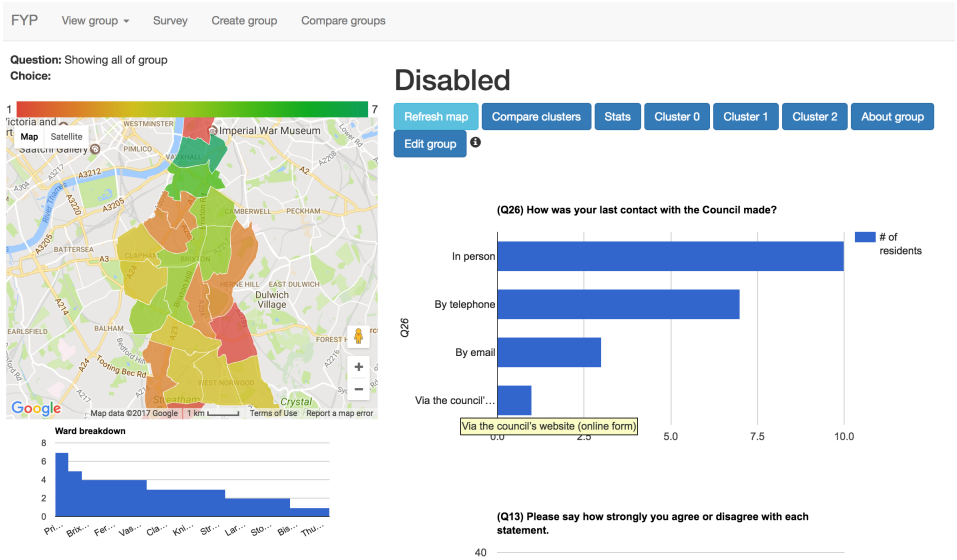

Figure 5.4: Screenshot of the group_detail page

Hovering over the map's wards will display a box containing the ward's name and value on the bottom left hand side of the map (see figure 5.5).

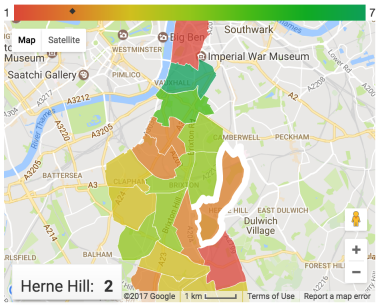

Figure 5.5: Screenshot of hovering over a ward on the Google Map

5.3.4 Visualization the Comparison of Clusters

A group's data, the whole survey data and clusters' data were visualized together for each question in a similar method to the visualization of a single group. To differentiate between each of the clusters' data for each question, the cluster id was included into the chart array. This was kept hidden when the array is displayed on the chart, but it is used to know which cluster is clicked when the user wants to display a particular variable on the map.

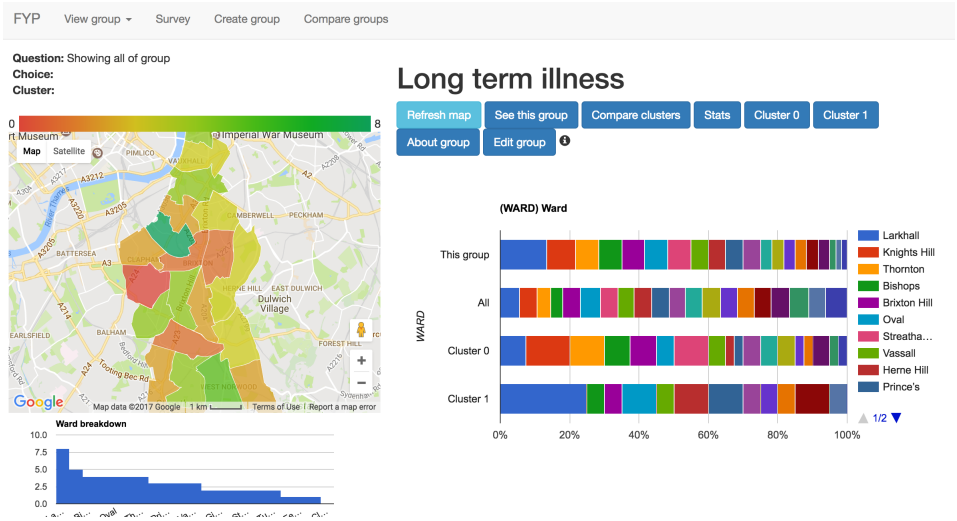

Figure 5.6: Screenshot of the cluster_compare page

All charts are stacked bar charts. The click to display on map feature is applicable to these charts as well.

5.3.5 Comparison of groups

On the right-hand side of the page (see figure 5.7), the charts generated from the All group with the same method as in a group_detail page. However, the left-hand side of the page is a bar chart of the group's percent difference from the whole survey population. As with a group's

Detail page, clicking a piece of data on a chart will instigate a change in the left-hand side bar chart.

The percent difference is calculated using the equation found in the Design chapter (see section 4.6)

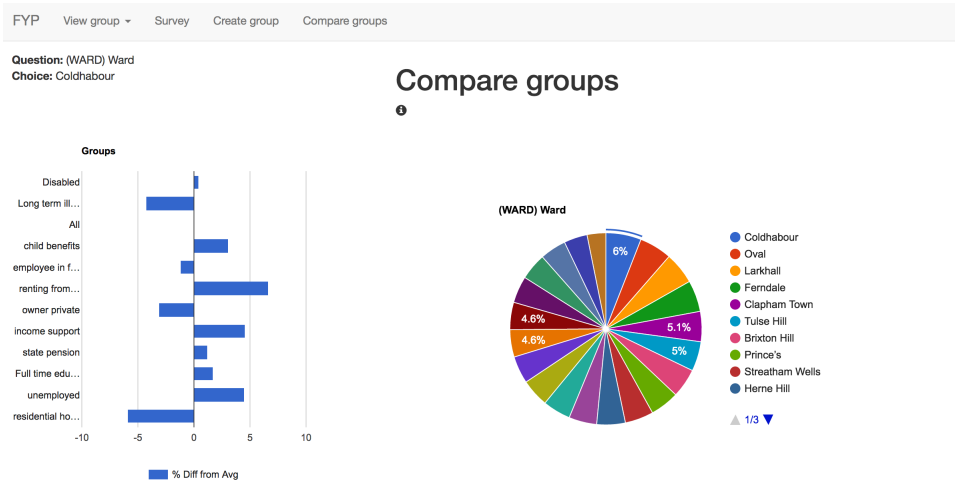

Figure 5.7: Screenshot of the group_compare page

5.3.6 Clustering algorithm

Each group was clustered according to the number of clusters set by the user in the Stats page. The default number of clusters is 1. K-modes clustering (citation) was used since it is more effective than k-means clustering to cluster categorical data, which the survey data is mainly composed of.

The results of clustering are stored in the database. The Subcluster Django model created the database table with fields: serial (i.e. serial of row in Survey data), group (i.e. Cluster model) and cluster (i.e. id of cluster). Should the number of clusters be changed by the user, the Subcluster records of the group will be deleted and replaced. If the clusters are queried, then the system will query the Subcluster database table so that the clusters remain the same.

5.3.7 Resource question page

Similar to the clusters_compare page and implemented with a similar Python operation, the survey question page displays more than one group in the same chart.

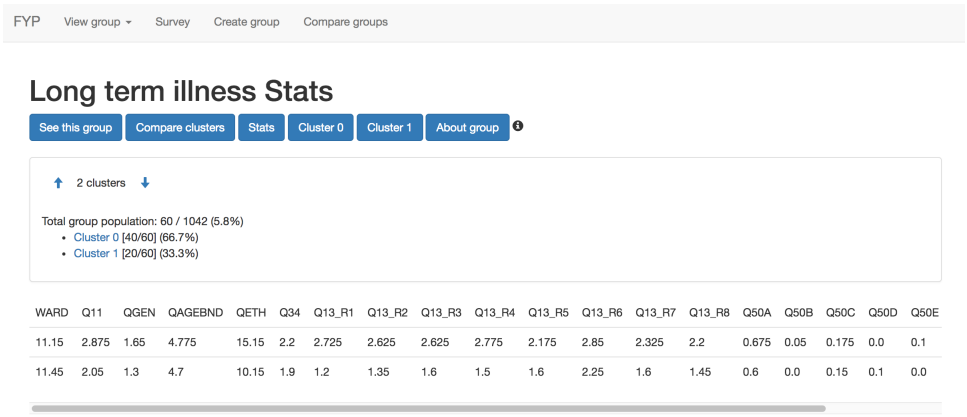

Figure 5.8: Screenshot of the cluster_stats page

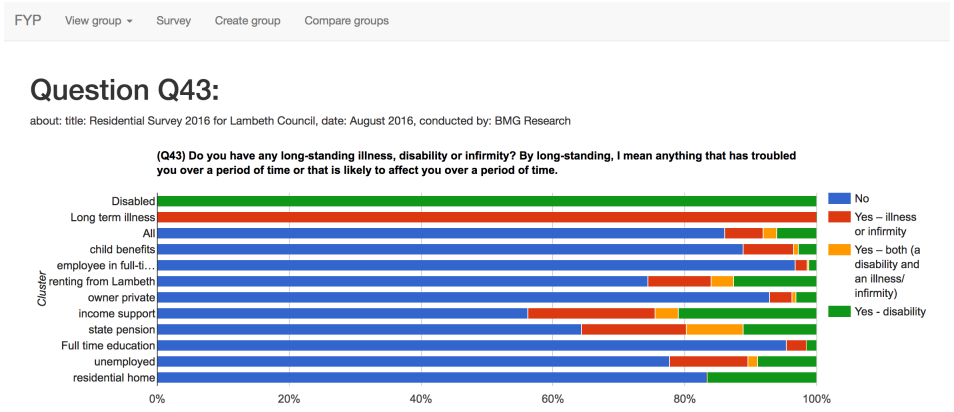

Figure 5.9: Screenshot of the resource_datafield page

5.4 Testing

To be continued

Chapter 6

Professional and Ethical Issues

The piece of software produced is a combination of my work and the work of Third Parties, which include Open Source libraries such as Django (citation), Graphos (citation), Google Maps (citation), Google Charts (citation), Kmodes (citation), and others (citation). These libraries were utilized to take advantage of existing work however, they were adapted to the purposes of this project. Relevant libraries and sources are cited in both source code and the description in the implementation to give credit to third party content. The main data used, the 2016 Lambeth Residential Survey, keeps its participants anonymous and according Lambeth has no security issues in terms of publicizing the data. The software places a resident under an anonymous serial, no name of any sort is saved nor a specific address. The software maintains the anonymity of the location of their residence at the Ward level instead of a more specific post code level.

The clustering of the survey data used a third-party software. Though sensitive information such as race and sex is included as variables in which to cluster, as far as I know the clustering algorithm used do not discriminate on those variables.

During the evaluation stage, a demonstration of the software was conducted to a group of 4 people from the policy department Lambeth Council to gain their feedback. The demonstration and the acceptance of their feedback was done respectfully as they have given honest criticisms of my work.

Chapter 7

Evaluation

The evaluation of the software involved the comparison between the design to the implementation. Since it is a visualization project, an independent evaluation was taken from a demonstration to some of the potential users from the policy department in Lambeth Council. The method and results of the evaluation is described below.

7.1 Demonstration to Lambeth Council

The demonstration took place in the offices of Lambeth where I presented each page of the web application. The potential of the visualization to aid the formation of strategies were discussed as well as the limitations and improvements of each feature. The requirements were given to them to compare to the actual software as well. The limitations are described below and the solutions to these limitations are in section âĂę of the Conclusion chapter.

They have expressed that it has met the objectives overall and that it has the potential given that there would be more features and an option to include data sets to name a few.

The subsections below summarizes the discussions for the visualization pages and the data used.

7.1.1 Data used

The project focuses on a prototype to explore the possibilities of visualizing a population rather than creating a complete tool with usable data. The current survey data could be used as it is to facilitate the initial analysis of the population though the visualization should be taken with caution.

They have expressed that the usefulness of the tool relies on the quality and amount of data used. The tool may not be usable at its current state since the survey data is not representative. The current data set used only surveyed 1024 residents of a council of more than 300,000 residents. It is also uses data from only 1 year. The software does not let the user change which annual residential survey to view nor does it utilize past residential surveys.

7.1.2 Group_detail page

This page gave much information as it not only gave data for each question, but also displaying a piece of data on the map showed where those residents live. This may help in identifying areas which need the most help.

In terms of the usefulness of the map, dividing it into wards limits the targeting of a smaller area since wards occupy a large area. Generalizing a ward to behave a certain way may ignore the minorities within the area. Using a smaller geographical division such as the Office of National Statistics' (ONS) output area or super output area which cover areas smaller than a ward and is included in the current data can identify more specific areas.

Due to the smaller data set, should there only be one resident in a ward with a given survey question, the identification of a more specific area, such as ONS' output area, would enable the user to identify if the user. For example, a ward with 1 resident in the disabled group could potentially be a person living in a senior home. This could give insights to people living in senior homes rather than a valid representation of disabled people in that ward.

The map's color range was misunderstood as red meant danger and green meant good. The choice of colors could have been constrained to shades of one color.

7.1.3 Group_compare page

This page gave a good indication on which questions are more applicable for which group. For example, the visualization shows more people in long-term illness and acquires child benefits are more likely to say they are not paid the London Living Wage. This could potentially help in identifying which resources should be given to which groups of people.

The information is limited to the percentage of a piece of data in relation to the data of a question. The comparison between groups and clusters are also in the form of percentages. Other statistical data such as mean, mode, median and standard deviation could be integrated which results in deeper analysis.

7.1.4 Clusters pages

Clustering in a statistical sense, needs to be explained to the users. Upon demonstrating the tool to the council workers, the concept of clustering needed to be explained. The visualization of the clusters as seen in cluster_detail or cluster_compare could only be understood given that clustering. The cluster_stats page was confusing as the table of means were still in the survey code. There needed to be English translations of the code supplemented. This was fixed after the demonstration where hovering over the question will show the English translations.

7.1.5 Resource_datafield page

They were interested in this page as it has the potential to confirm or debunk some of their expectations of some groups. For example it confirmed that the ethnicities of people living in residential homes were from English, Caribbean or African descent. Another application could be how they would approach some of the groups using the questions regarding the medium of contact with the council or how they access the internet.

7.2 Project evaluation

Though the tool was inspired by the problems faced by a council based on the interview with a Lambeth council worker, the requirements and design of the tool was up to me through research into other visualization applications. Therefore, the end tool may not be suitable for them. However, they did express that it has met the requirements I have set for the software. Regular contact during the whole process in terms of feedback of the design and usability, and more specific data sets would have produced a more usable product for their needs.

In terms of scalability, where the data set may be more than 1042 rows, the use of Pandas Dataframe may not be suitable as it could run into performance issues. Using a database should have been used to prevent this.

Chapter 8

Conclusion and Future Work

This aim of this project was to create a tool that will aid the formation of strategies to help segments of a population by a local council. The end product was a Django web application which gives different perspectives through different visualizations of a single data set. The tool's usefulness depends on the quality of the data and whether the data represents the population well. According to some members of the policy department in Lambeth Council, it has potential to be used should there be more thought put in what data sets are to be used.

The tool may have applications for other local governments dealing who would like to get to know a population more and explore what sorts issues are present within segments of the population. Outside of local government, this could be used for anyone looking to do customer segmentation such as businesses who want to tailor their services to the surrounding residents.

8.1 Future work

- The tool is implemented as a Django web application and should there be improvements, the tool could be packaged as such. This could enable the developers in local councils or potential users of the tool to implement it themselves using their own data sets and explore their population accordingly. Some pages could be turned into reusable Django Views class that can input a data set and output a page with the visualizations which could be used in other Django applications.

- The user could benefit with more flexibility in terms of which questions are to be included as the only certain questions are shown and which data set is to be used. Including other years' data sets could show whether a segment is improving or not. The way in which

the questions are visualized could also be improved by letting the user chose what type of graph the data should be represented. Giving other options such as a column chart or box plot to name a few will give more insight into the distribution of the data.

- Utilizing different sizes of geographical divisions such as ONS' output areas or super output areas instead of wards could be beneficial in targeting more specific areas.
- In other applications of the tool, the data may be more sensitive and require security features.
- The comparison of the results on two different questions as graphs or maps could be added as another page. This would enable the user to compare questions rather than groups. It could highlight correlations between clusters as well. For example, a map showing the residents who claim child benefits and another map showing the residents who live in a council flat might show the same trends and could be interpreted as a correlation between the two.
- The use of a more performance database such as MongoDB to store the data set could be implemented in future iterations of the program to increase scalability for data sets larger than one those rows. More statistical information such as mean, median, mode and standard deviation which standard knowledge in local government will provide more insight to the distribution of the data.

References

[1] 3. customer segmentation and profiling.

[2] Customer segmentation.

[3] Customer segmentation.

[4] Developing a customer classfication tool: Guidance document for local authorities.

[5] Kent & medway interactive guide.

[6] Brigitte Boden, Roman Haag, and Thomas Seidl. Detecting and exploring clusters in attributed graphs: a plugin for the gephi platform. In *Proceedings of the 22nd ACM international conference on Conference on information & knowledge management*, CIKM '13, pages 2505–2508. ACM.

[7] Drew Conway. Data science through the lens of social science. In *Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '14, pages 1520–1520. ACM.

[8] Anna Gogolou, Marialena Kyriakidi, and Yannis Ioannidis. Data exploration: A roll call of all user-data interaction functionality. In *Proceedings of the Third International Workshop on Exploratory Search in Databases and the Web*, ExploreDB '16, pages 31–33. ACM.

[9] Alvaro Graves and James Hendler. Visualization tools for open government data. In *Proceedings of the 14th Annual International Conference on Digital Government Research*, dg.o '13, pages 136–145. ACM.

Appendix A

Extra Information

A.1 Tables, proofs, graphs, test cases, ...

The appendices contain information that is peripheral to the main body of the report. Information typically included in the Appendix are things like tables, proofs, graphs, test cases or any other material that would break up the theme of the text if it appeared in the body of the report. It is necessary to include your source code listings in an appendix that is separate from the body of your written report (see the information on Program Listings below).

Appendix B

User Guide

B.1 Instructions

You must provide an adequate user guide for your software. The guide should provide easily understood instructions on how to use your software. A particularly useful approach is to treat the user guide as a walk-through of a typical session, or set of sessions, which collectively display all of the features of your package. Technical details of how the package works are rarely required. Keep the guide concise and simple. The extensive use of diagrams, illustrating the package in action, can often be particularly helpful. The user guide is sometimes included as a chapter in the main body of the report, but is often better included in an appendix to the main report.

Appendix C

Source Code

C.1 Instructions

Complete source code listings must be submitted as an appendix to the report. The project source codes are usually spread out over several files/units. You should try to help the reader to navigate through your source code by providing a "table of contents" (titles of these files/units and one line descriptions). The first page of the program listings folder must contain the following statement certifying the work as your own: "I verify that I am the sole author of the programs contained in this folder, except where explicitly stated to the contrary". Your (typed) signature and the date should follow this statement.

All work on programs must stop once the code is submitted to KEATS. You are required to keep safely several copies of this version of the program and you must use one of these copies in the project examination. Your examiners may ask to see the last-modified dates of your program files, and may ask you to demonstrate that the program files you use in the project examination are identical to the program files you have uploaded to KEATS. Any attempt to demonstrate code that is not included in your submitted source listings is an attempt to cheat; any such attempt will be reported to the KCL Misconduct Committee.

You may find it easier to firstly generate a PDF of your source code using a text editor and then merge it to the end of your report. There are many free tools available that allow you to merge PDF files.