Algorithms for Map Generation and Spatial Data Visualization in LIFE

MASTERARBEIT

Ying-Chi Lin
Leipzig, March 2016

Supervisors:
Dr. Anika Groß
Institut für Informatik
Abteilung Datenbanken
Prof. Dr. Erhard Rahm

Dr. Toralf Kirsten
Leipzig Research Center for Civilization Diseases (LIFE)
Acknowledgements

This thesis could not be finished without the help, kindness and knowledge input from my supervisors Dr. Anika Groß and Dr. Toralf Kirsten. Their support and inspiration on shaping the structure of the LIFE Spatial Data Visualization System and during the implementation phase are really appreciated. Great ideas and valuable suggestions have helped me to improve the writing of the thesis. Thanks also go to the office mates, Mandy Vogel, Matthias Rühle, Alex Kiel, Jonas Wagner and Diana Pietzner, for their readiness to help others, at any time! I am very happy to know them and really enjoyed the time I was in the office. Last, but not least, my gratitude also goes to my lovely families and friends, both in Taiwan and Germany, for their never ending supports.
Abstract

The goal of this master thesis is to construct a software system, named the LIFE Spatial Data Visualization System (LIFE-SDVS), to automatically visualize the data obtained in the LIFE project spatially. LIFE stands for the Leipzig Research Centre for Civilization Diseases. It is part of the Medical Faculty of the University of Leipzig and conducts a large medical research project focusing on civilization diseases in the Leipzig population [86]. Currently, more than 20,000 participants have joined this population-based cohort study. The analyses in LIFE have been mostly limited to non-spatial aspects. To integrate geographical facet into the findings, a spatial visualization tool is necessary. Hence, LIFE-SDVS, an automatic map visualization tool wrapped in an interactive web interface, is constructed. LIFE-SDVS is conceptualized with a three-layered architecture: data source, functionalities and spatial visualization layers. The implementation of LIFE-SDVS was achieved by two software components: an independent, self-contained R package lifemap and the LIFE Shiny Application. The package lifemap enables the automatic spatial visualization of statistics on the map of Leipzig and to the extent of the authors knowledge, is the first R package to achieve boundary labeling for maps. The package lifemap also contains two self-developed algorithms. The Label Positioning Algorithm was constructed to find good positions within each region on a map for placing labels, statistical graphics and as starting points for boundary label leaders. The Label Alignment Algorithm solves the leader intersection problem of boundary labeling.

However, to use the plotting functions in lifemap, the users need to have basic knowledge of R and it is a tedious job to manually input the argument values whenever changes on the maps are necessary. An interactive Shiny web application, the LIFE Shiny Application, is therefore built to create a user friendly data exploration and map generation tool. LIFE Shiny Application is capable of obtaining experimental data directly from the LIFE database at runtime. Additionally, a data preprocessing unit can transform the raw data into the format needed for spatial visualization. On the LIFE Shiny Application user interface, users can specify the data to display, including what data to be fetched from database and which part of the data shall be visualized, by using the filter functions provided. Many map features are also available to improve the aesthetic presentation of the maps. The resulting maps can also be downloaded for further usage in scientific publications or reports. Two use cases using LIFE hand grip strength and body mass index data demonstrate the functionalities of LIFE-SDVS. The current LIFE-SDVS sets a foundation for the spatial visualization of LIFE data. Suggestions on adding further functionalities into the future version are also provided.
CONTENTS

6.2 The LSA data access unit and tab Data .............................................. 48
6.3 Functionalities on tab Filter ................................................................. 51
6.4 Map functionalities for continuous data .............................................. 52
6.5 Map functionalities for categorical data .............................................. 53

7 Use Cases and Analysis ........................................................................ 55
7.1 Use case for continuous data: hand grip strength ................................ 55
7.1.1 Visualization of hand grip strength data ....................................... 56
7.1.2 Age-standardization of hand grip strength data ............................. 59
7.2 Use case for categorical data: body mass index ................................... 61

8 Conclusions and future work ................................................................. 66
8.1 Future work on LIFE-SDVS ................................................................. 68

A Appendix ................................................................................................. 69
B Appendix ................................................................................................. 71
List of Tables .............................................................................................. 73
List of Figures ............................................................................................ 74
References ................................................................................................. 76
1 | Introduction

In this master thesis, a software system, named the LIFE Spatial Data Visualization System (LIFE-SDVS), is conceptualized and implemented. LIFE stands for the Leipzig Research Centre for Civilization Diseases. It is part of the Medical Faculty of the University of Leipzig and conducts a large medical research project focusing on civilization diseases in the Leipzig population [86]. LIFE-SDVS aims to automatically visualize the assessment results in the LIFE project on the map of Leipzig. The first part of the thesis (Chapter 2) introduces the objectives and scopes of the LIFE project and also the data collected. To visualize these data, the architecture of LIFE-SDVS is developed and the implementation of this software system is described in the second part of the thesis (Chapters 3 - 6). The last part of the thesis demonstrates some use cases of LIFE-SDVS (Chapter 7).

1.1 Data visualization

Vision is the most dominant sensory system in the human brain [24]. Roughly 20-30% of the total surface area of the cerebral cortex is involved in visual processing ([104], [132]). The human brain can grasp the meaning of many data points faster while displayed in charts or graphs rather than reading through long list of spreadsheets or pages of reports. Especially in this era where massive amount of data are produced every day, detecting patterns and finding meaningful information becomes very challenging. Using visualization tools can lead to greater insights in less time. Therefore, data visualization plays an important role in the big data analysis pipeline (e.g. [15], [116]).

Data visualization is the science of visual representation of data [60]. The main goal of data visualization is, according to Friedman, “to communicate information clearly and effectively through graphical means” [59]. Michael Friendly [60] further classifies data visualization into two main foci: (1) statistical graphics: applies to any domain in which graphical methods are employed in the service of statistical analysis and (2) thematic cartography: primarily concerned with representation constrained to a spatial domain. The earliest visualization examples arose in geometric diagrams, in tables of the positions of stars, and in the making of maps to aid in navigation and exploration [60].
1.2 Spatial data visualization in medical research

Medical geography or health geography is an area of medical research or a subdiscipline of geography that incorporates the application of geographical information to the study of health, disease, and health care ([70], [95]). The interlink between medical researches and geography can be found in the literature of several ancient civilizations, including China, Greece, and India [97]. In his book “On Air, Water and Places” [67], Hippocrates (ca. 460 – ca. 370 BC) is probably the first to describe the relationship between the inhabitant’s health and the geographical characteristics of a place [55]. The term medical geography was first used among French physicians in the 18th century [26]. The first disease map can be dated back to 1792, a manuscript map by German physician Leonhard Ludwig Finke [25].

Perhaps the most famous example of using cartographic applications as a tool in early medical science is the research of the English physician John Snow (1813 – 1858). In his book On the Mode of Communication of Cholera, Snow reported his research on cholera in London. He used statistics to demonstrate that the degree of contamination of the drinking water corresponded to the number of cholera deaths. He plotted the clusters of cholera cases around the public well pumps during the epidemic of 1854 (Figure 1.1). His map pinpointed the pump on Broad Street as the most likely source of the outbreak. The pump caused 500 cholera deaths within ten days after contamination with sewage from a nearby pipe. Snow persuaded the local council to remove the pump handle and afterwards the number of cholera patients reduced rapidly. Snow’s study is regarded as the founding event of the science of epidemiology and he is considered as one of the fathers of modern epidemiology.

Figure 1.1: Spot-map by John Snow showing the clusters of cholera cases in London outbreak in 1854. In the enlarged part, the location of the water pump on Broad Street is indicated. The map originates from [122].
With the advancing of computer technology, software systems such as geographical information systems (GIS) transformed medical geography into a more analytic discipline during the latter half of the 20th Century [97]. One of the early notable medical GIS software is the Geographic Analysis Machine (GAM) [102]. Developed by Openshaw and his colleges in 1987, GAM was used to investigate the clustering effects of Leukemia and other cancers. The system contained not only a GIS for geographical display but also a spatial hypothesis generator and a significance assessment procedure. In addition to epidemiological applications, GIS has been applied in health services sectors ([20], [43], [63], [94], [114]). The public health departments, research organizations of hospitals, medical centers, and health insurance organizations use the information provided by these spatial software systems, for example, to determine where and when to intervene, to improve the quality of care, or to increase accessibility of service ([100]). Beyond analyzing geographical relevant information, GIS enables policymakers to better assess potential risk factors and prevent diseases ([38]).

Health atlas helps to explore general patterns of diseases that can be used to generate aetiologic clues. These maps can also be used to identify specific locations where changes in health policy need to be made. The U.S. Cancer Atlas helped the researchers to uncover the link between snuff dipping and oral cancer [139] and the relationship between shipyard asbestos exposure and lung cancer [34]. The Atlas of United States Mortality produced by the U.S. Center for Disease Control and Prevention (CDC), is the first to show all leading causes of death by race and sex in all regions of the U.S. at a higher resolution ([106], [107]). New mixed effects models and a weighted head-banging algorithm are used to improve the maps showing spatial trends more clearly. World Health Organization (WHO) also features similar health data for the world with its Global Health Atlas [11]. The users can explore data about the distribution of diseases such as HIV/AIDS or influenza in different countries to find patterns of transmission. Community Health Map is a web application that allows users to interact with the visualized health care data at the county level in the U.S. [123]. In this paper, a detailed review on similar projects is given. The Institute for Health Metrics and Evaluation (IHME) at the University of Washington is an independent global health research center. Their webpages on data visualization and infographics also include a few heath atlas [12]. However, none of these systems has direct access to databases and therefore the actuality of the data displayed is not endorsed. Moreover, they focus mostly on presenting maps but not function as a map generation tool.

1.3 Aims and outlines of this thesis

Since the assessment programs in the LIFE project started in 2011, a large amount of data has been collected. The amount of the data collected is still increasing with the ongoing experiments and the follow up studies. To efficiently explore the patterns in these data and to gain further medical insights, the support of data visualization software systems is necessary. In this thesis, the spatial data visualization software system LIFE-SDVS is developed. By using LIFE-SDVS, spatial related questions in the scope of the LIFE study, e.g. which regions of Leipzig have a high prevalence of diabetes, can be answered. Furthermore, LIFE-SDVS
can also act as a foundation for the new research project, the Leipzig Health Atlas. This project is funded within the "Integrative Datensemantik in der Systemmedizin" (i:DSem) program of the German Federal Ministry of Education and Research (BMBF) [3]. The Leipzig Health Atlas intends to break the barrier that most of the medical research data are only used by scientists for publications but not for other purposes, e.g., as an information tool for the users in the clinics. To achieve this, the LIFE data will be integrated into an IT-platform and various applications for the analysis will be provided [9].

LIFE-SDVS is a spatial data visualization software system equipped with an automatic map visualization tool and wrapped in an interactive web interface. To cope with the large amount of data available in the LIFE project, LIFE-SDVS provides direct access to the LIFE database so that the users can freely choose the data to be visualized. The data is further processed by the automated data preprocessing unit. The automatic map visualization tool generates maps to present the LIFE results with a spatial aspect. LIFE-SDVS goes beyond the function of only showing the results for exploration, the software system is also an interactive map generation tool. With just a few clicks and settings on the web interface, the LIFE scientists are able to produce customizable maps for research publications or public health reports.

The following chapter introduces more details of the LIFE project and some related work for building LIFE-SDVS. Chapter 3 presents the design of the architecture and the software components of LIFE-SDVS. Chapter 4 explains the data preprocessing unit. The automatic map visualization tool is presented in Chapter 5. Chapter 6 illustrates the interactive web application. Some use cases for LIFE-SDVS are demonstrated in Chapter 7. Conclusion and future works are stated in Chapter 8.
2 | Background and Related Work

This chapter firstly introduces the LIFE project and its three main studies: the LIFE-Adult-Study, the LIFE-Child-Study and the LIFE-Heart-Study. Section 2.2 explains the data management in LIFE, the potential data to be visualized spatially and the current situation of spatial data visualization in LIFE. While comparing the statistics among different regions of Leipzig, the underlying cofounder structures of different regions shall also be considered. Section 2.3 explains such so-called standardization methods. The last section (Section 2.4) of this chapter gives an overview on relevant map labeling research.

2.1 The LIFE project

The LIFE project is the first and largest population-based cohort study of this kind in an urban population in the eastern part of Germany [86]. In population-based cohort studies, researchers take a sample or even the entirety of a defined population for longitudinal assessments [128]. Unlike randomized control trials (RCTs), which is considered as gold-standard for determining the efficacy of clinical interventions, population-based cohort studies play an important role in scientific discovery and for narrowing the scope for RCTs [124]. Population-based cohort studies often aim to search for exposure-outcome relations, e.g. to investigate the influence of genetic, environmental, social and lifestyle factors on health and diseases. The findings of population-based studies should not be limited to the individuals included in the study but be generalizable to the whole population addressed in the study hypothesis [85].

With initial fundings from the European Union (34.1 million Euro) and the Free State of Saxony (5.5 million Euro), LIFE is the largest scientific project of the Saxon excellence initiative [86]. LIFE is part of the Medical Faculty of the University of Leipzig. Additionally, researchers of the Max Planck Institute for Human Cognitive and Brain Sciences Leipzig and the Leipzig Heart Centre are also involved. The LIFE project includes three subset studies: the LIFE-Adult-Study, the LIFE-Child-Study and the LIFE-Heart-Study. All study participants are inhabitants of Leipzig, a city with population size of approximately 550,000, mostly of central European descent.

The objective of the LIFE-Adult-Study is to investigate prevalences, early onset markers, genetic predispositions, and the role of lifestyle factors on major civilization diseases, such as metabolic and vascular diseases, brain malfunction, depression, sleep disorders, retinal and...
optic nerve degeneration, cognitive impairment, allergies and vigilance dysregulation [86]. Between August 2011 and November 2014, approximately 10,000 randomly selected participants from Leipzig attended the baseline examination. The main age group was of 40 to 79 years. A subset of 400 participants aged 18 to 39 years were also recruited (Figure 2.1). All participants in the main age group accomplished an extensive core assessment program (5-6 h) including questionnaires, structured interviews, physical examinations, and biospecimen collection. Two additional assessment programs (3-4 h each) including deeper cognitive testing, diagnostic interviews for depression, brain magnetic resonance imaging, and electroencephalography were taken by participants over 60 years. To investigate if body fat distribution is associated with functional traits of the brain and traits of eating behavior, a subcohort of 1200 participants aged 18-79 underwent abdominal and brain MRI-scans (magnetic resonance imaging, MRI) [86].

The aim of the LIFE-Child-Study is to assess how metabolic, environmental and genetic factors affect the health from fetal life to adulthood [109]. In addition to monitoring the normal growth, development and health, the LIFE-Child-Study also focuses on diseases such as childhood obesity, atopy and mental health problems. The study includes a neonate subcohort, the LIFE Child BIRTH cohort. In this cohort, extensive assessments are undertaken in pregnant women and their offspring from 24th week of gestation to 12 month of age. Afterwards the child is integrated into the LIFE Child HEALTH cohort, in which also other children and their families are recruited. The LIFE Child HEALTH cohort is a population-based cohort
of children and adolescents in an age range from 3-18 years. Additional specific assessments are carried out for each of the focused groups: the LIFE Child OBESITY cohort, the LIFE Child DEPRESSION cohort and the LIFE Child ATOPY cohort [109]. The LIFE-Child-Study is designed as a longitudinal cohort study and the participants are followed annually over a period of ten years.

The scientists in the LIFE-Heart-Study aim to assess biochemical and molecular biomarkers and their ability to evaluate the presence and severity of coronary artery disease (CAD) and to predict the future course of disease [31]. A biobank and database of patients with different stages of CAD are established for investigating the clinical, metabolic, cellular and genetic factors of cardiovascular diseases. The clinical cohort of approximately 7,000 heart patients consists of a sub-cohort with patients undergoing first-time diagnostic coronary angiography and a sub-cohort of acute myocardial infarction survivors. The study is one of the largest fully genotyped studies worldwide with angiographically assessed coronary patients [86]. Moreover, follow-ups at 5-year intervals will provide information about major cardiac clinical events of the study participants.

Since 2014 the LIFE research centre has joined the German National Cohort (GNC) as one of the 18 regional study sites [64]. The recruiting of new participants has started and aims to set a nationwide cohort of 200,000 individuals aged 20–69 years. The objective of the GNC is to investigate the causes for the development of major chronic diseases such as cancer, diabetes, cardio-vascular diseases and psychiatric diseases. The duration of the GNC is planned for 25–30 years.

2.2 Database and spatial data visualization in LIFE

The data management in LIFE consists of several commercial and self-developed software systems running in a shared network [86]. Most of the assessment and administrative data are stored in an ORACLE® relational database (Oracle Corporation, Redwood Shores, CA, USA). Participant data such as their identity, pseudonymisation, and appointments are organized separately from the scientific data in a dedicated participant management system. CryoLab, a self-developed laboratory information system (LIMS), manages the workflow of all biospecimen, such as their collection, labeling, processing, aliquoting, distribution and storage. Currently there are more than 700 assessments (i.e., investigations) including interviews, questionnaires, physical examinations, such as for anthropometry, EKG, MRT, and laboratory analyses of biospecimen applied in the different studies of LIFE. All the data of these assessments and their analysis results are integrated in a central research database. The ‘LIFE Investigation Ontology’ (LIO), developed by LIFE researchers Dr. T. Kirsten and A. Kiel, describes and classifies the entities in the research database and their relationships semantically [80]. A set of ontology-based tools is implemented to ease the data retrieval process in this large project [130]. The LIO is integrated into an ontological framework so that the scientists can utilize LIO to formulate queries over the scientific data. These ontology-based queries are transformed into SQL queries and the retrieved data are stored as project-specific
2.3. AGE-STANDARDIZATION OF THE STATISTICS

Many attribute measurements taken for clinical research or public health studies are gender- and/or age-dependent, i.e. the physical or mental traits differ between male and female and often hold a trend through aging. For example, hand grip strength of men is on average stronger than that of women. Furthermore, hand grip strength generally remains stable for man with age between 20 and 55 and starts to decrease from the age of 55 years ([93] Figure 1). A similar trend is seen in women of different age groups ([93] Figure 2), though the ampli-
2.3. AGE-STANDARDIZATION OF THE STATISTICS

The prevalence of diabetes in different areas of Leipzig:

- Schönefeld-Ost: 18.6%
- Althen-Kleinpösna: 2%
- Schleußig: 6.2%
- Probstheida: 7.2%
- Neustadt-Neuschönefeld: 7.6%
- Sellerhausen-Stünz: 17.5%
- Marienbrunn: 16.8%
- Meusdorf: 16.1%
- Möckern: 15.7%
- Wahren: 12.8%
- Lützschena-Stahmeln: 13.4%
- Lindenthal: 13.3%
- Böhlitz-Ehrenberg: 8.4%
- Leutzsch: 9.8%
- Grünau-Nord: 15.1%
- Grünau-Ost: 14.4%
- Eutritzsch: 15.1%
- Mockau-Nord: 15.4%
- Schönfeld-Ost: 18.6%
- Neustadt-Neuschönefeld: 7.6%
- Althen-Kleinpösna: 2%
- Probstheida: 7.2%

**Figure 2.2**: An example showing early spatial data visualization in LIFE.

The magnitude of decrease is not as large as that of the men (in men about 40% reduction, in women ca. 30%). Such statistics of attributes (e.g. arithmetic mean values, rates etc.) have to be statistically adjusted to remove the effect of population differences in age or gender structure when comparing between different study populations. **Standardization** is the statistical adjustment technique used in such instances. The factors influencing the statistics, such as the age or gender structure, are called **cofounders** [119].

There are two types of methods for the standardization: direct and indirect methods. For both method types, the standardized statistic is computed based on the statistic value of its stratum and a weight. Strata are subgroups to which individuals are aggregated by a certain attribute, for example, into three age groups (three strata). The calculation for the direct method is the following:

**DIRECT METHOD**

\[
{s}_{\text{standardized}} = \sum_{i=1}^{k} \left( \frac{w_i^{(\text{ref})}}{w_i^{(\text{total})}} \times s_i^{(\text{stu})} \right)
\]

(2.1)

where \( w_i^{(\text{ref})} = \frac{\text{number of individuals in stratum } i \text{ of the reference population}}{\text{total number of individuals in reference population}} \)

\( k \) is the total number of strata. The term \( w_i^{(\text{ref})} \) is the weight for stratum \( i \) of the reference population. Value \( s_i^{(\text{stu})} \) is the statistic in the study population for stratum \( i \).
In the indirect method, the weight for stratum \( i \) is derived from the proportion of stratum \( i \) in the study population and the statistic is obtained from the reference population:

**INDIRECT METHOD**

\[
\hat{s}_{\text{expected}} = \frac{1}{\sum_{i=1}^{k} \left( w_{i}^{(\text{stu})} \times s_{i}^{(\text{ref})} \right)}
\]  

where

\[
w_{i}^{(\text{stu})} = \frac{\text{number of individuals in stratum } i \text{ of the study population}}{\text{total number of individuals in study population}}
\]

The main conceptual difference between both methods is that in the direct method the statistic is obtained from the study population and the weight is from the reference population. On the other hand, in the indirect method study population provides the weight and the reference population provides the statistic in focus \[120]. The indirect method is used in cases where no stratum-specific measurement has been taken or in the instances where stratum-specific numbers are too small (such as populations in a single industrial plant or a small city) so that the stratum-specific estimates are too susceptible to random variability \[120]. Both cases do not occur in the regions of Leipzig. In consequence, the direct standardization method is chosen for LIFE-SDVS.

### 2.4 Map labeling algorithms

Map labeling or label placement is a fundamental task in cartography. Labeling quality depends on many factors and reflects human visual perception and experience \( [39], [75] \). The cartographers Imhof \( [71], [72] \) and Yoeli \( [146] \) have studied extensively on map labeling and proposed some basic rules of a good labeling: i) no overlaps of a label with other labels or other objects, ii) each label can be easily identified with exactly one graphical feature iii) a label must be placed in the most preferred position. However, these rules are descriptive and it is difficult to quantify these characteristics and come up with an appropriate definition of an objective function \[39]. The ACM Computational Geometry Task Force, a group of scientists, considered the automatic label placement problem as an important research area in their technical report "Applications challenges to computational geometry" \[16]\.

**Point-feature label placement**

Imhof classified the label-placement in cartography into three categories \( [71], [72] \): point-feature labeling (e.g. cities or mountain peaks), line-feature labeling (e.g. rivers or roads) and area-feature labeling (e.g. lakes or countries). Point-feature label placement (PFLP) as defined by Christensen et al. is "the problem of placing text labels adjacent to point features on a map or diagram so as to maximize legibility" \[39]. They further suggested the legibility of PFLP into an objective function considering following factors:

1. The amount of overlap between text labels and other labels or graphical features
determining the optimal placement of a label for an isolated line or area feature, the three placement tasks share a common

function.

The number of point features left unlabeled. (This criterion is pertinent only when point selection is incorporated into the

PFLP can be seen as a combinatorial optimization problem and has been proved as NP-hard even for very restricted labeling models ([54], [78], [90]). One example is when all text labels are rectangles of the same size, each text label has to be placed with one of its four corners at the point (as shown in Figure 2.3(a)) and all text labels have to be disjoint [54]. Therefore, for such problems the practical application of exact search algorithms is limited to problems with at most a few hundred points (if $P \neq NP$) ([44], [81], [126], [147]).

Various heuristic methods have been developed to tackle the PFLP. Christensen et al. [39] compared six heuristic algorithms including methods such as a greedy algorithm, an integer programming algorithm of Zoraster [147], a gradient-descent method by Hirsch [68] and a stochastic algorithm utilizing simulated annealing. The comparison shows that simulated annealing outperforms all other algorithms and is one of the easiest algorithms to implement. Different approaches of genetic algorithms (GA) have been proposed to solve the PFLP (e.g. [22], [110], [131], [134]). Algorithms applied on PFLP with the constraint that all point features must be labeled include a tabu search [144], a constructive genetic approach [51] and a fast algorithm for label placement [145]. Wagner et al. proposed a two phase algorithm for the situation that certain labels are allowed to be omitted and the objective is to place as many labels as possible with no overlaps [135]. Doddi et al. developed constant-factor polynomial-time approximation algorithms to solve the generalized map-labeling problem, i.e. labels are not limited to be rectangular but can also be elliptical, any restriction on their orientation is removed and allowing the point feature can be placed anywhere on the boundary of its label region [47]. An extensive Map-Labeling bibliography is maintained by Wolff and Strijk [141].

**Line-feature label placement (Edge Label Placement)**

Edge Label Placement (ELP) or line-feature labeling is the problem of assigning text labels to lines (edges) such that the association of the labels to their corresponding edges is unambiguous [73]. In ELP, a label can touch the edge it is labeling but it should not overlap any other graphical feature in a drawing [74]. For example, labels like A, B and D in Figure 2.4 are preferable and a label like C, which overlaps its associated edge, can be acceptable with
some appropriate cost assigned to it. The ELP problem is proved to be NP-hard ([36], [73]).

Figure 2.4: Labeling space of an edge. Figure adapted from [74].

Area-feature label placement
The general accepted practice of area-feature labeling in cartography are: i) the labels must be inside the boundaries of the area and ii) the labels must span the entire area and conform to its shape ([19], [56], [57], [58], [72], [108], [133], [146]). Figure 2.5 shows a resulting map of an automatic area-feature labeling [19]. Since an area name should span the area, it is necessary to find a shape description or baseline of the area. It was firstly done by the so-called skeleton method [96]. To reduce the strong influence of minor boundary irregularities, Ramer [112] proposed a boundary-approximation algorithm to smooth the boundary first. Edmonson and Christensen [52] label areas by using random methods and their scoring function uses the centroid as an ideal position for the label text. Pinto and Freeman argued that [108] it did not seem possible to devise an algorithm to solve area-feature labeling properly in every instance. Therefore, they developed a method that deploys a feedback approach consisting of two components: i) the Placement Generator generates a large set of potential text placements for a given region and ii) the Placement Evaluator evaluated these candidates. The evaluation criteria follows the main guidelines for spread-out area feature labeling such as longer, spread-out placement is preferred over a shorter one, a good placement shall conform to the shape of the area and larger clearance from the boundary is preferred.

Dörschlag et al. [50] proposed a vector-orientated algorithm for placing not only text but also objects such as diagrams in areas of a map without violating the boundary of the areas. A bounding box is a rectangle that outlines the label text or the object. The aim of the algorithm is to determine a point in the area of a map and the center point of the bounding box should be placed on this determined point. The algorithm includes the following three steps:

1. Erosion of an area: this step reduces the potential candidate area where the center point of the bounding box should ideally be placed. The candidate area is the original area subtracting an inward buffer (Figure 2.6). The buffer is obtained by moving the bounding box along the original area border. If the candidate area consists of several disjunct regions, one of these regions should be chosen heuristically.

2. Calculate the skeleton: this step reduces the candidate area to a 1-dimensional skeleton. To choose candidate points on the skeleton, methods such as to choose the mid-points of the skeleton edges or to select the skeleton vertices with a degree of at least three are suggested. However, exact prescription is missing.
2.4. MAP LABELING ALGORITHMS

A PROGRAM FOR AUTOMATIC NAME PLACEMENT

Figure 3: Example of automatic area-feature placement.

Post-processing Editor

Although it is intended that the annotation system be completely automatic, it is clear that some provisions must be included for interactive editing of the result. The purpose of such editing is to improve the appearance of the map, to correct data errors, and to correct possible mistakes made by the system.

Currently, there are point annotation algorithms have been implemented on a PRiME-750 computer with a program written in RATFOR, a FORTRAN preprocessor.

3. Select final point: this step uses a heuristic with a scoring function to determine one point of the skeleton as the final point for placing the center point of the bounding box.

Boundary labeling

Boundary labeling, though commonly applied in practice, was first studied by Bekos et al. in 2004 [29] (later also published as [30]). In their boundary labeling model, the features are enclosed within a rectangle, each label is connected to its associated feature through a polygonal line called leader, and no two leaders intersect. The boundary labeling problem can be defined as follows. Given is a set \( P = p_1, \ldots, p_n \) of points and an axis-parallel rectangle \( R \) that contains \( P \). Each point, or site, \( p_i \) is associated with an axis-parallel rectangular open label, where open means that the border of the label is not part of the label. The labels have to be placed and connected to their corresponding sites by leaders, such that i) no two labels intersect, ii) no two leaders intersect, iii) the labels lie outside \( R \) but touch \( R \), and iv) each leader lies inside \( R \).

Typically additional requirements or restrictions are made to the boundary labeling problem. For example, only certain restricted types of leaders may be allowed (examples of leader
2.4. MAP LABELING ALGORITHMS

![Figure 2.6: Example of the erosion step of the algorithm by Dörschlag et al.](image)

Figure 2.6: Example of the erosion step of the algorithm by Dörschlag et al. [50] (a) the origin area and the bounding box (b) the erosion process (c) the result. The figure is adapted directly from the paper.

Types see Figure 2.7. The number of line segments of each leader might be restricted and also the type the line segments might be restricted. Other requirements consider the length of the leaders, e.g. the total length of all leaders should be minimal or the total number of bends of all leaders should be minimal. There might also be a fixed point – called label port - on the side of the label that touches \( R \) and where the corresponding leader has to be connect to the label, e.g. the middle point of the edge.

![Figure 2.7: Examples of leader types in boundary labeling.](image)

Figure 2.7: Examples of leader types in boundary labeling. The figure is adapted from [30].

Many boundary labeling algorithms are proposed for different leader types and different sides of \( R \). It was shown in [30] that the boundary labeling problem where each leader has only one line segment, i.e. the leader is a straight line, can be solved in time \( O(n \log n) \). The same time bound holds also for the corresponding one-sided boundary labeling problem [30]. A recent overview on boundary labeling problems where leaders have more than one line-segment is in [28]. Nöllenburg et al. [99] considered the scenario of one-sided boundary labeling where the user can interactively select regions of interest by zooming and panning the map. Problem of two-sided boundary labeling with adjacent sides is dealt in [79]. Huang et al. [69] extended boundary labeling with flexible label positions so that labels do not necessarily form only one single stack and the two ends of label stacks can extend over the sides of the map. They showed that when using nonuniform-size labels, almost all of the total leader length minimization problems are NP-complete.

Map labeling requirements in LIFE-SDVS

The city map in LIFE-SDVS is based on the data set provided by the Leipzig city government. The data set contains coordinates of the border points of the administrative regions at two lev-
els: Ortsteile and Stadtbezirke. These given regions are used to present the results obtained by the LIFE assessments. For the visual representation, short labels such as the Ortsteile IDs can be used to denote the regions. Furthermore, it is also desirable to show statistical graphics or symbols in each region of Leipzig. Inspired by Figure 2.2, the long label texts that can not be fitted into the regions can be placed outside of the map area. For such boundary labeling, the leaders connect a point in each region and extend to the margins of the map. Consequently, good positions within each region for placing the short labels, graphics and as starting point of leaders for the boundary labeling are necessary. These positions are called labeling positions in this thesis. To find such good labeling positions within regions is one of the tasks of the label placement problem in LIFE spatial data visualization.

The applicability of map labeling algorithms in LIFE-SDVS
Some fundamental differences restrict the application of the classical map labeling algorithms on the current stage of LIFE-SDVS. The point-feature label placement algorithms are designed to place text labels adjacent to a given set of point features while the LIFE data set contains given region border points and good positions for placing labels are yet to be found. The area-feature label placement algorithms generally aim to place label text along the baseline within the associated area but LIFE-SDVS does not limit to show text within each region but also graphics. The algorithm proposed by Dörschlag et al. [50] has similar objectives, i.e. to find a good labeling position of an area in map. However, the authors gave only the conceptual descriptions on how the algorithm was structured but not a detailed depiction. Thus, some missing information makes the application of the algorithm difficult. Furthermore, there is also no indication of how good the algorithm works. To solve the boundary labeling problem in LIFE-SDVS, a similar idea to that of Bekos et al. [30] is applied.
This chapter explains the conceptual design of LIFE-SDVS. In the first section the architecture of LIFE-SDVS is introduced. The different software components and their relationship to the architecture are described in the second section.

3.1 Architecture of LIFE-SDVS

LIFE-SDVS comprises a three-layered architecture: data source, functionalities and spatial visualization (Figure 3.1). The data source is the LIFE research database containing the assessment results of the LIFE-Adult-Study and the LIFE-Child-Study. The functionalities consists of two main components: data access and data preprocessing. Data access is for accessing the LIFE database by SQL queries to obtain the raw data. The raw data is then preprocessed using functionalities including filters, statistical aggregation and standardization. The output statistics of data preprocessing are used for the spatial visualization layer. In this layer, the statistics can be displayed on the map of Leipzig. LIFE-SDVS is finally presented as a web application user interface backed by a web application server. Within the server, an independent tool is responsible for the map visualization. Through this system, the users can obtain the data directly from the LIFE database and produce maps interactively. Additionally, the web application user interface can also function as a web presentation for displaying the LIFE research results in the different regions of Leipzig on a map.

The programming language R and the Shiny web application framework are chosen to build the LIFE Spatial Data Visualization System. LIFE-SDVS shall be capable of applying statistical analysis, plotting graphics, visualizing spatial data on maps and contains an interactive web application. All these tasks can be implemented very well by using R and Shiny.

R is a free software environment for statistical computing and graphics [4]. It is one of the most comprehensive statistical analysis packages available [138]. Guy Harrison argued that while the commercial statistical software, e.g. SPSS or SAS, “price tags increased, while the focus on business intelligence did not always align with academic desires... R arguably represents the most accessible and feature-rich set of statistical routines available” [66]. There are several other advantages of R. R is backed by the CRAN package repository which currently...
3.2 Software components of LIFE-SDVS

The three-layered architecture of LIFE-SDVS is implemented with two software components: the visualization package named lifemap and the LIFE Shiny Application (Figure 3.2). The visualization package lifemap is an independent, self-contained R package that produces the maps. The design of gathering all the map visualization functions as its own package
3.2. SOFTWARE COMPONENTS OF LIFE-SDVS

Figure 3.2: The relationship of three-layered architecture and the software components of LIFE Spatial Data Visualization System. The map visualization tool in the spatial visualization layer is implemented in the R package `lifemap`. The connectivity to the LIFE database, the data preprocessing layer and the web application are realized in the LIFE Shiny Application. Reducing the complexity of the LIFE Shiny Application and makes the software maintenance simpler. The LIFE Shiny Application builds the skeleton of LIFE-SDVS. It contains the data access and data preprocessing components in the functionalities layer and also the web application user interface and the web application server in the spatial visualization layer.

The package `lifemap` turns the spatial data into maps of Leipzig and adds additional features onto these maps. The main features include displaying aggregated statistics on each region of Leipzig, placing labels in good positions within each region and providing customizable boundary labeling function. To realize these features, the `lifemap` package supplies two plotting functions: `plot_continuous` and `plot_categorical`. The former visualizes the continuous data of the assessments in LIFE database and the latter is for visualizing categorical data. To enhance the aesthetics of the maps produced, the `lifemap` package is equipped with two self-developed labeling algorithms: the Label Positioning Algorithm (LPA) and the Label Alignment Algorithm (LAA). LPA aims to find good positions within each region on the map for placing labels or statistical graphics (see Section 5.1). The boundary labeling function places the labels of the regions outside of the map and uses leaders to connect the regions with their labels. These leaders might intersect with each other when the labels are
not placed in a proper order. The objective of LAA is to find a proper ordering of the boundary labels, i.e. to solve the intersection problem of the leaders (Section 5.2.2).

The second software component of LIFE-SDVS is the LIFE Shiny Application (LSA). The application acts as a web presentation platform for the maps and at the same time is also an interactive map generation tool itself (details see Chapter 6). The LSA server imports the lifemap package and uses its plotting functions to visualize the maps. The generated maps are then displayed on the LSA user interface. By interacting with the LSA user interface, the users can choose which data to be displayed on the map, use the filter functions to select only a subset of the data to visualize and to customize many map features to generate maps for publications or for reports. These functions involve the interactions of the LSA data access unit, the LSA data preprocessing unit and the spatial visualization in the LSA user interface and the LSA server (Figure 3.2).

The following chapter introduces what kind of data types in the LIFE research database are visualized and the functionalities of the LSA data preprocessing unit. Chapter 5 presents the map visualization package lifemap, its plotting functions and the two labeling algorithms: LPA and LAA. Chapter 6 describes the structure of the LIFE Shiny Application, mainly focusing on the LSA data access unit, the LSA user interface and the LSA server.
4 | Data Preprocessing

In this chapter, the LSA data preprocessing unit, a component of the functionalities layer of LIFE-SDVS is introduced (Figure 3.2). The main functionalities in this unit are filter, statistics and standardization. After the raw data has been obtained directly from the LIFE research database by SQL queries, users can use filters to specify which data subset to be visualized. Moreover, depending on the data types, different statistics can be applied for the visualization on map and for displaying in a data table on the LSA user interface. At the end of the chapter, the age structures of the reference population are determined for the age-standardization of different gender groups.

Data types to visualize
LIFE-SDVS aims to visualize two data types from the LIFE database: continuous data and categorical data. Continuous data are numeric values such as the number of sampled individuals of an assessment, the age of a proband or the measurements of blood pressure. Categorical data are the data of categorical nature, e.g. the gender of probands, or the quantitative data that have been converted into that form. For example, the body mass index (BMI) data in LIFE database are available both in continuous form (index values between 16 and 58) and in categorical form (category 1 to 4, 1 is underweight and 4 is obese). Users can select two region levels to show the statistics on the map: nine Stadtbezirke or 63 Ortsteile.

The two different data types are visualized differently on map of Leipzig. Various cognitive research recommend that choropleth map style is good for map showing patterns ([84], [89], [105], [107], [137]). Hence for the continuous data, choropleth map style is used to display different ranges of means, medians and absolute frequencies (number of sampled individuals) in different regions of Leipzig. Age-standardized means can also be displayed for three gender categories: both genders, male and female (Section 4.3). For the categorical data, pie charts or bar charts for each region display the absolute frequencies of different attribute categories.

Data preprocessing pipelines
To visualize different types of maps for the two data types, different data preprocessing processes are needed. Figure 4.1 shows the data preprocessing pipelines for continuous data and categorical data. The same filter functions are applicable for both data types. Different statistical approaches are applied to different data types. The details are described in Section 4.1 and Section 4.2 for continuous and categorical data, respectively. The age-standardization
method introduced in Section 2.3 can be applied to the means of continuous data to obtain the standardized means (Figure 4.1(b)).

Filters
The user can utilize the filter function to visualize particular sub groups of data. Among the many assessment data stored in the LIFE database, three attributes are common in all the analysis. These are (1) gender, (2) age, and (3) absolute frequency. Hence, these three attributes are chosen for the filter function on both continuous data and categorical data. The gender filter specifies if the data of only male, only female or of both genders shall be visualized. Currently, LIFE-SDVS focuses on the visualization on data of LIFE-Adult-Study, hence the age filter contains three age groups: (18,40], (40,60], and (60,80+]. If in a region only limited number of participants attended an assessment, the statistics obtained based on this small sample is not representative. Therefore, the absolute frequency filter allows the user to set a minimum absolute frequency of a region. In the visualization of continuous data, the regions with sample size less than the threshold are colored in a specific color to indicate there is too little or no data available in the region. Similarly, the pie charts or bar charts are absent in such regions in the visualization of categorical data.

4.1 Data preprocessing of continuous data

Format of raw data
Each row of the raw data contains the information of one proband \( D = (R, G, A, V) \), where \( R \) is in which region the proband lives, e.g. the Ortsteil ID, \( G \) and \( A \) are the gender and age of the proband, respectively, and \( V \) is the value of the continuous data to be visualized. Table 4.1 shows an example from hand grip strength data. Columns region, gender and age are needed for the filter functions.
4.1. DATA PREPROCESSING OF CONTINUOUS DATA

Table 4.1: Format of the raw data for the continuous data visualization in LIFE-SDVS. Here shows the partial data of the hand grip strength collected in LIFE. Column region is the Ortsteile ID of the proband. Column value is the measured hand grip strength value.

<table>
<thead>
<tr>
<th>region</th>
<th>gender</th>
<th>age</th>
<th>value</th>
</tr>
</thead>
<tbody>
<tr>
<td>90</td>
<td>2</td>
<td>63</td>
<td>30</td>
</tr>
<tr>
<td>11</td>
<td>1</td>
<td>63</td>
<td>46</td>
</tr>
<tr>
<td>13</td>
<td>2</td>
<td>35</td>
<td>38</td>
</tr>
<tr>
<td>52</td>
<td>2</td>
<td>28</td>
<td>26</td>
</tr>
<tr>
<td>52</td>
<td>2</td>
<td>69</td>
<td>21</td>
</tr>
</tbody>
</table>

Statistics aggregated by filter conditions

Various statistics are displayed on the LSA user interface for continuous data. In addition to the visualization of the mean, median and absolute frequency on the map, a data table is also shown. Each row in the data table displays the information of a region on the map. The information comprises the region IDs, absolute frequency, mean, standard deviation, standard error, median and 95% confidence intervals. If the gender filter is applied and only a subgroup of the gender is selected to display (i.e. only male or only female), a column containing the selected gender is also shown in the data table. Similarly, a specific age group is displayed (i.e. (18,40], (40,60] or (60,80+]) if the age filter is applied.

To obtain the statistics for the map visualization and the data table, the raw data are aggregated with the following function:

```r
get_group_data_con <- function(data, in_gender = "all", in_age = "all", groupvars = "region")
{
  # select the subset of the raw data and aggregate into groups
  group <- data %>% filter(gender %in% in_gender, age_class %in% in_age) %>%
    group_by(.dots = groupvars)
  # apply summarise function to each group data
  dplyr::summarise(group, num = sum(num), mean = mean(value), sdV = sd(value),
                   median = median(value),
                   q1V = quantile(value, 0.25), q3V = quantile(value, 0.75))
}
```

In the function arguments at line 1, in_gender and in_age are specified by the gender and age filter, respectively. Aggregation by region (i.e. groupvars = "region") returns the statistics for each region of Leipzig. The filter() (line 4), group_by() (line 5) and summarise() (line 7) functions are supplied by R package dplyr. The functions within dplyr::summarise() returns the absolute frequency (num), mean of the values (mean) and standard deviation (sdV) of each group data. Further functions quantile(value, 0.25) and quantile(value, 0.75) (line 9) return the first and third quantiles, respectively, and the function median() returns the second quantile. The function mutate of package dplyr enables rowwise computation of data frame in R. The following code shows the application of mutate function to obtain the standard error (se), the lower bound (lci) and upper bound (uci) of the 95% confidence interval of each region (row):
where the `inputData` is the output of the previous function `get_group_data_con`.

### 4.2 Data preprocessing of categorical data

#### Format of raw data

The raw data format for categorical data visualization is similar to that for continuous data. Each row of the raw data contains the information of one proband $D = (R, G, A, C)$, where $R$, $G$ and $A$ are the region, gender and age of the proband, respectively, and $C$ contains the information to which category the proband belongs (Table 4.2).

Table 4.2: Format of the raw data for the categorical data visualization in LIFE-SDVS. Shown are only the partial data of the BMI collected in LIFE. Column `region` is the Ortsteile ID of the proband. Column `category` is the BMI category the proband belongs to.

<table>
<thead>
<tr>
<th>region</th>
<th>gender</th>
<th>age</th>
<th>category</th>
</tr>
</thead>
<tbody>
<tr>
<td>95</td>
<td>2</td>
<td>44</td>
<td>2</td>
</tr>
<tr>
<td>40</td>
<td>2</td>
<td>41</td>
<td>2</td>
</tr>
<tr>
<td>15</td>
<td>2</td>
<td>68</td>
<td>3</td>
</tr>
<tr>
<td>64</td>
<td>2</td>
<td>58</td>
<td>4</td>
</tr>
<tr>
<td>21</td>
<td>1</td>
<td>48</td>
<td>2</td>
</tr>
</tbody>
</table>

The values in the `category` column in the raw data are coded numerically. The text denotation for these codes are defined in the so-called `Derivattabelle` in the LIFE research database (an example of a Derivattabelle is attached in Appendix B). The codes and their corresponding text denotation are extracted as a metadata view. The legends of the maps are the obtained from this view.

#### Statistics aggregated by filter conditions

Absolute frequencies of each category of an assessment (e.g. BMI) are displayed in pie charts or bar charts within each region of Leipzig. The corresponding program code is as follows:

```r
get_group_data_cat <- function(data, in_gender = "all", in_age = "all", 
                                groupvars = "region", in_category = "all")
{
  # select the subset of the raw data and aggregate into groups
  group <- data %>% filter(gender %in% in_gender, age_class %in% in_age, 
                           category %in% in_category) %>% group_by(.dots = groupvars)
  # apply summarise function to each group data
  dplyr::summarise(group, num = sum(num))
}
```
Similar to the function `get_group_data_con` for continuous data, the raw data can be filtered by specific gender and age groups and aggregated by region IDs (line 5-6). Moreover, an aggregation of the absolute frequency of each category is applied (line 6). In contrast to continuous data, `dplyr::summarise()` function only applied on the absolute frequency (num in line 8):

### 4.3 Standardization

For the visualization of continuous data, the means of a certain assessment (e.g. hand grip strength) in each region can be shown on the map. A mean calculated so far is the average of the measured values of the sampled participants within each region. These means have to be standardized if the age structures within each region are different (see Section 2.3 for more detail). Table 4.3 shows examples of the age structures of the population with age > 20 years in three selected Leipzig Ortsteile. Half of the population with age between 20 and 80+ in Ortsteil Schleußig consists of individuals less than 40 years old. On the contrary, there are almost 60% of the inhabitants in Grünau-Ost are with age over 60 years. Comparing these two Ortsteile, the age structure in Eutritzsch area is more evenly distributed, with the age group (20,40] having only 8% more population than the other age groups. The age structures of male and female show a similar trend as the both_gender structure in Schleußig and Grünau-Ost. However, in Eutritzsch, the second largest age group of male population is (40,60] (34%) and the third is the age group (60, 80+] (30.3%). In contrast, the second largest age group of female is (60, 80+] (34.9%), while the age group (40,60] consists only 28.3% of the Eutritzsch female population. This example shows that it is important not only applying standardization on age structure based on both_gender, different age structure of different genders shall also be applied.

<table>
<thead>
<tr>
<th>age group</th>
<th>Schleußig (OT50)</th>
<th>Grünau-Ost (OT61)</th>
<th>Eutritzsch (OT93)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>both_gender</td>
<td>male</td>
<td>female</td>
</tr>
<tr>
<td>(20,40]</td>
<td>52.5</td>
<td>52.2</td>
<td>52.7</td>
</tr>
<tr>
<td>(40,60]</td>
<td>30.2</td>
<td>32.2</td>
<td>28.3</td>
</tr>
<tr>
<td>(60,80+]</td>
<td>17.3</td>
<td>15.6</td>
<td>19.0</td>
</tr>
<tr>
<td>population</td>
<td>9680</td>
<td>4621</td>
<td>5059</td>
</tr>
</tbody>
</table>

The direct standardization method is carried out on the means of each region. To apply Equation 2.1, the age structure of the reference population is needed. Since the LIFE adult data are mostly collected between years 2011 and 2015, the population structure of city Leipzig in year 2013 is taken as the reference population. Table 4.4 shows the weights of each age stratum $w_{k}^{(ref)}$ in three gender categories derived from the Leipzig Ortsteilkatalog 2014 [125]. Among the adult population in Leipzig (age > 18) about 40% are under age 40 and each of the other two age groups takes up about 30%. This age distribution projects...
roughly that of the male adult population in Leipzig. However, female population is comprised of about 35% in age group (18-40] and (60-80+], and about 30% of age group (40-60].

In total, three age-adjusted standardization instances are applied, depending on which gender category is selected to be visualized with the gender filter. When both_gender are selected to be visualized on the map, the weights in the column both_gender in Table 4.4 are used. Similarly, the weights in column male or column female are applied when the data for each gender are displayed (Table 4.4).

**Table 4.4**: Proportions of each age group of Leipzig population in three gender categories. These values are used as $w_k^{(ref)}$ for each age stratum for the standardization of means. Data are derived from the population structure of Leipzig in year 2013 from Ortsteilkatalog 2014 [125].

<table>
<thead>
<tr>
<th>age group</th>
<th>both_gender</th>
<th>male</th>
<th>female</th>
</tr>
</thead>
<tbody>
<tr>
<td>(18,40]</td>
<td>0.39</td>
<td>0.40</td>
<td>0.37</td>
</tr>
<tr>
<td>(40,60]</td>
<td>0.30</td>
<td>0.32</td>
<td>0.28</td>
</tr>
<tr>
<td>(60,80+]</td>
<td>0.31</td>
<td>0.28</td>
<td>0.35</td>
</tr>
<tr>
<td>total population</td>
<td>539,348</td>
<td>262,336</td>
<td>277,012</td>
</tr>
</tbody>
</table>
5 | Visualization Package lifemap

This chapter demonstrates the main components of the visualization package lifemap in LIFE-SDVS, including two map labeling algorithms and two R plotting functions. Section 5.1 describes the Label Positioning Algorithm that aims to find good labeling positions within regions of a map. Section 5.2 explains on how continuous data is visualized in LIFE-SDVS. In addition to the internal labeling feature, boundary labeling is also realized as described in Section 5.2.1. The second labeling algorithm, Label Alignment Algorithm, is applied to solve the leader intersection problem in boundary labeling (see Section 5.2.2). Section 5.3 gives details the implementation of both map labeling algorithms in R. Section 5.4 demonstrates the visualization of categorical data.

5.1 Label Positioning Algorithm

For spatial data visualization on maps good labeling positions within each region are essential. The labeling positions are for placing labels or graphics inside regions. Moreover, they can be the starting points of leaders for boundary labeling. Geospatial data consist often of sets of points where each set defines a border of a region and the points are usually stored as coordinates (for an example see Figure 5.1). Even though there is no exact definition on an optimal label placement of a region, the following informal criteria can be used as practical guidelines for good labeling positions:

1. labeling position shall locate inside the region
2. labeling position shall keep good distance from any point of the region border so that labels do not overlap with the border excessively
3. labeling position shall roughly situate in the middle of the region for better visual effects

A common practice of setting labeling positions within each region is to use the centroids, i.e. the arithmetic mean position of all border points of the region:

\[ x_{\text{centroid}} = \frac{\sum_{i=1}^{k} x_i}{k} \]  (5.1)
5.1. LABEL POSITIONING ALGORITHM

Figure 5.1: Border of the Ortsteil Zentrum in city Leipzig. The geospatial data for this border consist of $k = 156$ points and each point is represented by a pair of $x$- and $y$-coordinates.

where $x_{\text{centroid}}$ is the $x$ coordinate of a centroid point, $x_i$ is the $x$ coordinate of the border point $i$ and the total number of border points of the region is $k$. The calculation for $y_{\text{centroid}}$ is analog. To compute the centroids of spatial data in R, the gCentroid function in the package rgeos can be used. Though not explicitly explained in the function description, a proof calculation confirms that the function does calculate the centroid points according to Equation 5.1. Figure 5.2 shows the resulting labeling positions denoted with IDs of each Ortsteil in Leipzig.

Figure 5.2: Map of Leipzig with IDs of Ortsteile positioned at the centroid points of each region.

The application of centroids as labeling points on the Leipzig map shows several violations of the criteria for good labeling positions. The label of Ortsteil 53 (OT53, this format is used to denote an Ortsteil along with its ID throughout this thesis) locates outside of its
area (filled with purple color). Many of the centroid points are too close to the border, such as OT10, OT51, OT72, OT81, OT93 (filled with blue color). Some centroid points (e.g., OT15, OT41, OT55, OT82) satisfy criteria (1) and (2), but do not situate close to the middle of the respective Ortsteil so that the visual effects still can be improved (filled with brown color).

To obtain better labeling positions, a new algorithm is proposed, namely the Label Positioning Algorithm (LPA). LPA aims to obtain good labeling positions for irregular shaped polygons. LPA is a generic algorithm, i.e. its application is not limited to the map of Leipzig. For a generic description, the following text utilizes the term ‘x-coordinate’ to represent longitude, the term ‘y-coordinate’ represents latitude and the term ‘polygon’ represents a geographical region.

LPA starts from finding the middle point of the polygon so that the labeling position locates roughly in the middle of the polygon. Further, with one- or two-step adjustments, another four candidate positions are generated. Among these five candidates, the point with largest minimum distance from the polygon border is selected as the labeling position for the polygon. Section [5.1.1] and Section [5.1.2] describe the concepts of LPA and Section [5.1.4] shows more details on the actual deployment of the algorithm.

5.1.1 Five candidates of Label Positioning Algorithm

The five candidates of LPA are:

1. the middle point $p_m$:
   \[
   \text{coordinates of } p_m : (x_{p_m}, y_{p_m}) = \left( \frac{x_{\text{max}} - x_{\text{min}}}{2}, \frac{y_{\text{max}} - y_{\text{min}}}{2} \right)
   \]  
   (5.2)

2. one-step x-adjusted point $p_{1x}$:
   \[
   \text{coordinates of } p_{1x} = \left( \frac{x_{p_{1x}} - x_{p_{1xl}}}{2}, y_{p_m} \right)
   \]  
   (5.3)

3. one-step y-adjusted point $p_{1y}$:
   \[
   \text{coordinates of } p_{1y} = \left( x_{p_m}, \frac{y_{p_{1y}} - y_{p_{1yb}}}{2} \right)
   \]  
   (5.4)

4. two-step x and then y-adjusted point $p_{2xy}$:
   \[
   \text{coordinates of } p_{2xy} = \left( \frac{x_{p_{2x}} - x_{p_{2xl}}}{2}, \frac{y_{p_{2y}} - y_{p_{2yb}}}{2} \right)
   \]  
   (5.5)

5. two-step y and then x-adjusted point $p_{2yx}$:
   \[
   \text{coordinates of } p_{2yx} = \left( \frac{x_{p_{2y}} - x_{p_{2yl}}}{2}, \frac{y_{p_{1y}} - y_{p_{1yb}}}{2} \right)
   \]  
   (5.6)
5.1. LABEL POSITIONING ALGORITHM

Figure 5.3: One-step position adjustments of LPA on OT75. (a) one-step x position adjustment starting from the middle point, \( p_m \). Point \( p_{1x} \) is the x adjusted position with x-coordinate is the mean of the x-coordinates of \( p_{lxl} \) and \( p_{rxr} \). In (b), the point in green color is the middle point \( p_m \). The point in yellow color is the one-step x adjusted position \( p_{1x} \) and the point in blue is the one-step y adjusted position \( p_{1y} \).

**Middle point:**

Equation 5.2 presents the formula used to calculate the ‘middle point’ \( p_m \) of a polygon. The notations \( x_{\max} \) and \( x_{\min} \) are the maximum and minimum values of x-coordinates among all polygon border points, respectively. The notations for the y-coordinates are analog. LPA uses \( p_m \) as the initial labeling position candidate.

**One-step adjustments:**

Starting from \( p_m \), two further candidates are generated by a so-called one step adjustment: each candidate keeps either the x- or y- coordinates of the middle point \( p_m \) (the starting point of the one-step adjustments) and the other corresponding coordinate position is obtained by taking again the middle position from the polygon border points with the given respective x- or y- coordinates. The aim of taking the mean of the two border points is to shift the labeling point away from the border. Figure 5.3(a) shows an example using OT75 (Burghausen-Rückmarsdorf) to illustrate the one-step x position adjustment. In the example, the middle point of the polygon \( p_m \) is quite close to the right border. Instead, it should be located in the middle at the same y-coordinate level, i.e. an adjustment of the x-coordinate is needed. To achieve this, LPA takes the right and left points of the polygon border which have the same y-coordinate (i.e. \( p_{1xr} \) and \( p_{1xl} \) in Figure 5.3(a)) and computes the mean of the x-coordinates of these two points (Equation 5.3). This new x-coordinate and the y-coordinate of the middle point \( p_m \) forms the second LPA candidate: \( p_{1x} \) (in the subscript, 1 for one step and \( x \) for x-adjusted). Correspondingly, the third labeling position candidate is obtained by the one-step y position adjustment in which the x-coordinate of \( p_m \) is used as the x-coordinate of this new point, \( p_{1y} \), and the y-coordinate of \( p_{1y} \) is the mean of the y-coordinate of the border point on the top, \( p_{1yt} \), and that of the border point at the bottom (Equation 5.4). Figure 5.3(b) shows the resulting two candidates from the one-step adjustments in OT75. Note that depending
Figure 5.4: Two-step position adjustments of LPA on OT75. (a) two-step x and then y position adjustment starting from the middle point, $p_m$. (b) shows candidates generated by two-step adjustments in OT75. The point in red color is $p_{2xy}$ and the point in purple color is $p_{2yz}$. The point in green color is the middle point $p_m$.

on the dataset, the border points for the x position adjustments (i.e. $p_{1xr}$ and $p_{1zl}$) might not have exactly the same y-coordinate as the middle point. Similarly, while locating $p_{1yt}$ and $p_{1yb}$ for one-step y adjustment, there might not exist two border points which have the same x-coordinates as the middle point. Section 5.1.4 describes more details on how these border points are located for the position adjustments.

Two-step adjustments:
The two-step position adjustments take the one-step x or y adjusted positions as starting points and a further one step y or x position adjustment, respectively, is carried out. Equation 5.5 shows the coordinates of the fourth candidate, $p_{2xy}$. In the subscript, 2 is for two steps and xy is for the fact that the adjustments are firstly done in x-coordinate and then in y-coordinate. The candidate $p_{2xy}$ uses the x-coordinate of the one-step x-adjusted point $p_{1x}$ as its own x-coordinate. Its y-coordinate is computed by taking the mean of the y-coordinate of the border point on the top ($p_{2yt}$) and the y-coordinate of the border point at the bottom ($p_{2yb}$) (Figure 5.4(a)). The fifth candidate $p_{2yz}$ is generated by firstly one-step y-adjusted point as in the third candidate and then a further x position adjustment (Equation 5.6). Both candidates obtained by two-step adjustments of OT75 are shown in Figure 5.4(b). As in one-step adjustments, depending on the dataset, the border points for position adjustments (i.e. $p_{2xl}$ and $p_{2xr}$ for x position adjustment, $p_{2yt}$ and $p_{2yb}$ for y position adjustment) might not have the same x or y-coordinates as the starting points. Details on the deployment of locating the border points are described in Section 5.1.4.
5.1. LABEL POSITIONING ALGORITHM

5.1.2 Selection of the best candidate in Label Positioning Algorithm

Generating five candidates raises the chance that a good labeling position can be found. The selection of the best among these five candidate positions, the criteria 2 (page 30) that "the labeling position shall keep good distance from any point of the region border so that labels do not overlap with the border excessively" is applied. Therefore, among the five position candidates generated by LPA (Equation 5.2 ~ Equation 5.6), the one that locates furthest away from any polygon border point is selected as the labeling position of LPA.

In the implementation, the Euclidean distances of each candidate to every polygon border point are calculated. The minimum Euclidean distances to the border points of each candidate are stored. Figure 5.5 depicts the five candidates and their minimum distances to the border of Ort10 Schönefeld-Abtnaundorf. Each line segment connects a candidate to the border point with which the minimum Euclidean distance is obtained. The actual shortest distance values are also shown for each candidate. The best labeling point among these five candidates is the one with the largest shortest distance, in this case, the two-step y and then x adjusted position (in purple).

5.1.3 Application of LPA on Leipzig Ortsteile

The resulting best labeling points generated by LPA for all Ortsteile of Leipzig are shown in Figure 5.6. Judged by the three criteria for good labeling position, the results are good.
Figure 5.6: Best labeling positions for each Ortsteil of Leipzig found by LPA. The numbers within each region are the Ortsteil IDs. The same coloring is used as in Figure 5.2 for easier comparison between the results using centroid and that found by LPA.

No label situates outside of its polygon. Only two of the 63 Ortsteile (OT63, OT70) having their labels overlap with their polygon borders, mainly due to the fact that these regions are too narrow to accommodate the labels. Most of the labels locate roughly in the middle of the polygons. Overall, the visualization effect is much improved from the results generated by centroids in Figure 5.2. Every of the five candidate types: the middle point and the four candidates from one-step and two-step adjustments, contribute to the final selected best labeling points in the case of Leipzig Ortsteile map (Table 5.1). The one-step y adjustment and the two-step first x and then y adjustment, both together consists approximately 2/3 of all solutions.

Table 5.1: Frequency (number of Ortsteile) where each candidate of LPA has been selected as best labeling position in Leipzig map.

<table>
<thead>
<tr>
<th>Candidates</th>
<th>middle point</th>
<th>one-step</th>
<th>two-step</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>x adjusted</td>
<td>y adjusted</td>
</tr>
<tr>
<td>Frequency of being selected as best labeling position</td>
<td>7</td>
<td>6</td>
<td>20</td>
</tr>
</tbody>
</table>
5.1. LABEL POSITIONING ALGORITHM

Table 5.2: First five rows of the polygon data of OT75 sorted in ascending order by column y_diff. Values in column p_id are the IDs of each polygon point. Column y_diff contains the difference between the y_coord of the polygon point and that of the middle point of OT75 ($y_{pm}[OT75] = 51.34522$). The point at the first row (colored in green) is the $p_{1xr}$ of OT75 for the one-step x position adjustment. The point at the other side of the polygon is the $p_{1xl}$ (colored in purple).

<table>
<thead>
<tr>
<th>p_id</th>
<th>x_coord</th>
<th>y_coord</th>
<th>y_diff</th>
</tr>
</thead>
<tbody>
<tr>
<td>13569</td>
<td>12.27484</td>
<td>51.34522</td>
<td>6.13192E-07</td>
</tr>
<tr>
<td>13576</td>
<td>12.28076</td>
<td>51.34522</td>
<td>5.30197E-06</td>
</tr>
<tr>
<td>13579</td>
<td>12.28381</td>
<td>51.34524</td>
<td>1.88210E-05</td>
</tr>
<tr>
<td>13803</td>
<td>12.24792</td>
<td>51.34525</td>
<td>2.44024E-05</td>
</tr>
<tr>
<td>13578</td>
<td>12.28370</td>
<td>51.34520</td>
<td>2.57770E-05</td>
</tr>
</tbody>
</table>

5.1.4 Locating border points for position adjustments

Every step of the position adjustment in LPA needs to find the two border points that have the same x- or y-coordinate as the starting point (e.g. Figure 5.3(a)). In other words, an x position adjustment in LPA needs to find the two border points that have the same y-coordinate as the starting point and a y position adjustment needs to find the two border points that have the same x-coordinate as the starting point. For example, the one-step x position adjustment starts from the middle point $p_{m}$ and the horizontal shift on the x position is given by the mean of the x-coordinates of the two border points $p_{1xr}$ and $p_{1xl}$ (see Figure 5.3(a)). Ideally, LPA locates these border points by screening through all the border points of the polygon and finds the two points having the same y-coordinate as the middle point $p_{m}$. However, such points might not exist in the available dataset. For this reason, these points are defined as the border points with their y-coordinates having minimum difference to the y-coordinate of the starting point and both locate on different sides of the starting point (e.g. Figure 5.3(a)). For the position adjustments with vertical shift (at y-axis), LPA finds the borders points with their x-coordinates having minimum difference to the x-coordinate of the starting point and both locate on either top or bottom side of the starting point (e.g. Figure 5.4(a)).

Table 5.2 shows the example of locating border points for the one-step x position adjustment in OT75. Every row is a polygon border point. The starting point for the position adjustment is the middle point $p_{m}[OT75]$ with the coordinates (12.27247, 51.34522). Column y_diff contains the differences between the y-coordinate of the border point and that of the middle point. The rows are sorted ascending by y_diff. The first row, the one with minimum y-coordinate difference to the starting point (with green color background), has larger x-coordinate than the starting point (12.27484 and 12.27247, respectively). Therefore, it is the $p_{1xr}$ on Figure 5.3(a). The $p_{1xl}$ (with purple color background) is the first border point that has smaller x-coordinate than the starting point, i.e. on the left hand side of the middle point.

Algorithm 5.1 shows the pseudo codes of LPA. For each polygon, the first candidate, the middle point, is generated using Equation 5.2 (line 1). For each of the four other candidates, a position adjustment consists of two functions: locateBorderPoints() and determineCoordinates(). The process to locateBorderPoints() (line 3) is described in Section 5.1.4. The coordinates
of each candidate are determined with Equations 5.3 - 5.6 (line 4). The best candidate position is selected as explained in Section 5.1.2 (line 6-9).

Algorithm 5.1 Label Positioning Algorithm (LPA)

**Input:** all border points of a polygon  
**Output:** the best labeling position  
1: generate first candidate, i.e., the middle point  
2: for each of the four other candidates do  
3:    locateBorderPoints()  
4:    determineCoordinates()  
5: end for  
6: for each candidate do  
7:    calculateMinimumDistanceToBorder()  
8: end for  
9: select the candidate with largest minimum distance to border as best labeling position

Further considerations of LPA

There are two further considerations when LPA is to apply in other regions on earth. The locating of the border points for position adjustments in LPA is based on all border points having positive x-coordinate and y-coordinate values (Section 5.1.4). This assumption does not affect the applicability of LPA in almost all the geographical regions on earth apart from the small areas extending over the meridian of 180° longitude, e.g. Iultinskiy rayon, Chukotskiy in Russia.

The other consideration is that the current LPA is not yet able to deal with cases where a starting point for position adjustment locates outside of the polygon. The locating of the border points for position adjustments (Section 5.1.4) functions when one border point locates on one side of the starting point and the other point on the other side of the starting point. When the starting point locates outside of the polygon, the two border points might locate on the same side of the starting point. This problem can possibly be solved by clustering the border points and setting thresholds to distinguish if a border point is on the left/right or top/bottom border of the polygon.

5.2 Visualization of continuous data in LIFE-SDVS

The continuous data in LIFE is visualized by the plotting function `plot_continuous` (Figure 5.7). Function `plot_continuous` is based on the R plotting system `ggplot2` [2]. The data for plotting map and labels are given in Figure 5.7 line 3. The statistics of LIFE data are shown on the map of Leipzig using choropleth map style. For example, coloring the regions with the means of hand grip strength of the population in Ortsteile of Leipzig results in the map shown in Figure 6.3. To color the regions, users can assign a data frame containing two columns: (1) the IDs of each region (2) the statistics of each region to the `fill_data` argument in `plot_continuous` (Figure 5.7 line 15). For example, a data frame consists
of the column region and column mean in Table 7.1. It is also possible to customize some aesthetic features such as label size, font and enabling legend (line 16). Setting alpha value as 0 removes the coloring of the regions (line 16).

```r
plot_continuous <- function {
  # data for plotting map and labels
  map_data, label_text, label_data,
  # assign regions to each external labeling group
  right_group=NULL, left_group=NULL, bottom_group=NULL,
  # assign positions of each labeling group
  right_top=51.43357, right_bottom=51.3519, left_alignment=12.52072,
  left_top=51.36014, left_bottom=51.27666, right_alignment=12.23666,
  bottom_right=12.44595, bottom_left=12.3366, top_alignment=51.2598,
  # set margin size to adjust the space for external labels
  margin_left=0.055, margin_right=0.02, margin_bottom=0,
  # set visibility of internal labels
  inLabel_off="no",
  # map features
  label_line_colour="blue", pal="Greens", fill_data=NULL, title=NULL,
  label_size=4, family="Helvetica", alpha=1, with_legend="yes"}{...}
```

**Figure 5.7:** Arguments in plotting function `plot_continuous` to enable boundary labeling and choropleth maps for continuous data in LIFE-SDVS.

The label positions found by LPA show good results on the map of Leipzig labeled with Ortsteil IDs (Figure 5.6). However, when long label texts are needed, e.g., replacing IDs with names of the Ortsteile, the outcome is not satisfactory (Figure 5.8). Boundary labeling, i.e. to place labels on the margins outside of the map, can solve this problem of overcrowding. Therefore, the following sections describe the realization of boundary labeling in LIFE-SDVS.

**Figure 5.8:** Map of Leipzig labeled with Ortsteil names at the best labeling positions found by LPA.
5.2. VISUALIZATION OF CONTINUOUS DATA IN LIFE-SDVS

5.2.1 Realization of boundary labeling in LIFE-SDVS

Inspired by the map showing the LIFE results of diabetes in different Ortsteile in Leipzig (Figure 2.2), boundary labeling in LIFE-SDVS shall fulfill the following conditions:

1. boundary labels are not only for labeling all the regions, but can be used to show selected information. Therefore, the plotting function shall allow the users to define which labels and information they wish to display
2. to leave the map a clear look, the internal labels can be set invisible
3. no labels at the top of the map but rather placing the labels on the right, left and bottom of the map
4. easy adjustment of the positions of the labels
5. automatically scaling the distance between labels

Boundary labeling in LIFE-SDVS is realized with the plotting function `plot_continuous` (Figure 5.7). The boundary labels are designed to be placed on the left, right and bottom margins around the map in three corresponding labeling groups: right_group, left_group and bottom_group (Figure 5.9(a)). Users can assign region IDs to each of these labeling groups (Figure 5.7 line 5). There are three arguments to set the position of each labeling group (Figure 5.7 lines 7-9). For example, the position of the right_group is defined by (1) right_top: the y-coordinate for the label on the top (2) right_bottom: the y-coordinate for the label on the bottom and (3) left_alignment: the x-coordinate for all the labels in the right_group (Figure 5.9(b)). The settings for the other two labeling groups are defined analogously. The plotting function `plot_continuous` sets equal distances between the labels within each labeling group automatically. The users can also adjust the margin size to fit the length of the label texts (line 11). Straight lines are chosen as the leader type for maps since they are simple, often used by professional graphic designers and show good performance with respect to user readability (comparison in recent discussion [27]).

The labeling positions generated by LPA are used as the starting points of the leaders which connect each polygon to its boundary label. To avoid the intersection of the leaders, the order of the labels within each boundary labeling group is important. One simple way is to determine the order of the boundary labels by the coordinates of the labeling positions within each polygon. For the labeling groups right_group and left_group, the internal labeling positions of the polygons are sorted descendant by their y-coordinates and the ones with larger y-coordinates are placed on the upper positions within their labeling groups. Similarly, the polygons of the bottom_group are sorted according to the x-coordinates of their internal labeling positions and the one with larger x-coordinates are placed on the righter positions. This intuitive way of ordering the labels within each labeling group does not work well and still results intersection problem (Figure 5.10). Hence, the Label Alignment Algorithm is proposed to solve the line segment intersection problem of boundary labels.
5.2. VISUALIZATION OF CONTINUOUS DATA IN LIFE-SDVS

Figure 5.9: Design of boundary labeling in LIFE-SDVS. Subfigure (a) shows the three labeling groups of boundary labels placed around map of Leipzig. Subfigure (b) shows the three arguments for the position setting of the right_group.

5.2.2 Label Alignment Algorithm

The order of the boundary labels within each labeling group determines if the leaders of the boundary labels intersect with each other. Which region’s label shall be assigned to which boundary labeling position is the following alignment problem called - Label Alignment Problem (LAP). Given a set of internal labeling positions \( P_i \) and a set of boundary labeling positions \( P_b \). The aim is to find an bijective assignment of the elements in \( P_i \) to the elements in \( P_b \) so that no line intersection occurs. The Label Alignment Algorithm (LAA) is designed to solve LAP.

Through observing, for example, the right_group in Figure 5.10, one can conclude that a line-line intersection occurs when the line segment of the boundary label above has larger slope than the line segment of the boundary label beneath. For example, the line segment of Ortsteil Mockau-Süd intersects with the line segment of Gohlis-Mitte because the first line segment is steeper than the second line. Hence, LAA solves the intersection problem by re-ordering the boundary labels based on their line segment slopes.

Algorithm 5.2 shows the pseudo codes of LAA. For each labeling group, the coordinates of the internal labeling positions and the coordinates of the boundary labeling positions are stored in a data frame. The boundary labeling positions are ordered from top to bottom by y-coordinates for the right_group and the left_group and ordered from right to left by x-coordinates for the bottom_group (sortBoundaryLabelPositions() at line 1). In each iteration, LAA takes the topmost or rightmost position and calculates the slopes from this boundary labeling position to all the internal labeling positions which are not yet been assigned (computeSlopes() at line 3). The internal labeling position with the minimum slope is then "assigned" to the boundary labeling position, i.e. its label is placed on this boundary labeling position (assignBoundaryLabel() at line 4). The ID of the assigned region, its internal labeling position and the paired boundary labeling position are removed from the input lists before the next iteration starts (removeAssignedPair() at line 5). Figure 5.11 shows the three boundary
5.2. VISUALIZATION OF CONTINUOUS DATA IN LIFE-SDVS

Figure 5.10: Map of Leipzig with boundary labeling plotted using function `plot_continuous` without the Label Alignment Algorithm.

Figure 5.11: Map of Leipzig with boundary labeling plotted using function `plot_continuous` with the application of the Label Alignment Algorithm. The same sets of labeling groups are used as in Figure 5.10.
labeling groups containing the same sets of Ortsteile as in Figure 5.10 but with application of LAA. The intersection problems of the right\_group and left\_group in Figure 5.10 are solved by LAA.

**Algorithm 5.2** Label Alignment Algorithm (LAA)

**Input:** a list of region IDs $RI = \{id_1, id_2, \ldots, id_k\}$, their labeling positions within regions $P_{in}$, a list of boundary labeling positions $P_{bo}$

**Output:** an alignment of region IDs in $RI$ to boundary labeling positions

1: sortBoundaryLabelPositions()  
2: for $iteration = 1 \rightarrow k$ do  
3: \hspace{1em} computeSlopes()  
4: \hspace{1em} assignBoundaryLabel()  
5: \hspace{1em} removeAssignedPair()  
6: end for

### 5.3 Implementation of map labeling algorithms in R

**Label Positioning Algorithm**

LPA is implemented as a stand alone R function because for each map it has to be applied only once. Figure 5.12 presents the actual input requirement and the partial output of the R function `find_label_point.r` to obtain the labeling positions of LPA. The users only need to input a data frame for the labels and a data frame containing border point coordinates of the polygons, LPA will return a data frame with six coordinate columns:

- $mid\_x = x_{p1x}$
- $mid\_y = y_{p1y}$
- $mid\_yx\_x = x_{p2yx}$
- $mid\_xy\_y = y_{p2xy}$
- $best\_x$: x-coordinate of the best candidate
- $best\_y$: y-coordinate of the best candidate

The resulting data frame for the Ortsteile of Leipzig is saved as `ot_label_points.rdata` and the resulting data frame for the Stadtbezirks is stored in `sbz_label_points.rdata`. These data are used for the two plotting functions in the package `lifemap`. Since LPA is a generic label position finding algorithm, users can input other `label\_data` and `polygon\_data` to get their own labeling positions.

**Label Alignment Algorithm**

The function `resolve_intersection` in the package `lifemap` implements LAA. The function takes two data frames as input: one stores the coordinates of the internal label points of a labeling group and the other contains the internal labeling positions for that labeling group.
5.4 Visualization of categorical data in LIFE-SDVS

The categorical type of data in LIFE are visualized as pie charts or bar charts using the plotting function `plot_categorical` in the package `lifemap`. The function `plot_categorical` is based on the `plot` function in R package `graphics`. The pie charts or bar charts within each region are placed on the labeling position found by LPA. The `barplot` function in the package `graphics` is used to plot each single bar chart. The coordinates of each bar chart within each polygon is assignable in the `subplot` function of the package `TeachingDemos`. The pie charts are implemented using the `floating.pie` function in the package `plotrix`. Also from the package `plotrix`, the function `color.legend` defines the legend features.

Figure 5.13 shows the arguments of `plot_categorical`. After specifying the map to be plotted, either "Ortsteil" or "Stadtbezirk", the respective polygon data and labeling positions
are assigned inside the function (line 3). The argument `inputDf` shall contain a data frame with three columns: (1) id of the regions (2) category names and (3) the absolute frequency of each category (example see Table 7.5). If the absolute frequency (the number of sampled individuals) of a region is less than the `filterValue`, the pie chart or bar chart of the region will not be displayed (line 3). The users can specify "bar" for bar charts and "pie" for pie charts in the argument `plotType` (line 5). Further settings for the legend features and parameters for the charts are also available in the `plot_categorical` (Figure 5.13 lines 7-13).

```r
plot_categorical <- function(  
  # specify the map to plot, input categorical data and a filter value  
  map="Ortsteil", inputDf=bmiDf, filterValue=0,  
  # specify bar chart or pie chart and a color list for the categories  
  plotType="bar", colList=NULL,  
  # legend features  
  legendText=NULL, textSize=0.8, textFont="Helvetica",  
  # legend position  
  legendPos_lx=12.35, legendPos_ly=51.24, legendPos_rx=12.357, legendPos_ry=51.27,  
  # set radius for pie charts  
  pieRadius=0.0045,  
  # set the x, y dimension of the bar charts  
  sizeX=0.3, sizeY=0.3){...}
```

**Figure 5.13:** Arguments in plotting function `plot_categorical` in LIFE-SDVS.

The two plotting functions: `plot_continuous` and `plot_categorical` in the package `lifemap` are the foundation of the visualization of the statistics of LIFE data on map of Leipzig. To enable users to produce maps more efficiently rather than inputing the values into these functions manually, an interactive data exploration and map generating tool, namely the LIFE Shiny Application is constructed (Chapter 6).
6 | LIFE Shiny Application

LIFE Shiny Application (LSA) builds the skeleton of LIFE-SDVS (Figure 3.1). LSA is capable of accessing data directly from the database and processing the raw data for the visualization. LSA enables the users to choose which data to be displayed on the map, use the filter functions to select only a subset of the data to display and to customize many map features to generate maps for publications or reports. As shown in Figure 3.2, LSA consists of four architecture components: the LSA user interface, the LSA server, the LSA data preprocessing unit and the LSA data access unit. The LSA data preprocessing unit has been introduced in Chapter 4. Hence, this chapter will focus mainly on the other three components.

Section 6.1 gives firstly an overview on the software composition of LIFE Shiny Application. The following sections introduce the functionalities of the application, in the same order as the tabs are available on the LSA user interface. There are two user interfaces in LSA, one for continuous data and the other for categorical data (see the top navigation tabs on the screenshot of Figure 6.3). Section 6.2 explains the LSA data access unit and the data input options on the tab Data. Section 6.3 presents the filter options on tab Filter of the user interfaces. Since different map styles are used to visualize continuous data and categorical data, the functionalities for the map features also differ. The map functionalities for continuous and categorical data are demonstrated, respectively, in Section 6.4 and Section 6.5.

6.1 Software composition of LIFE Shiny Application

Shiny is the web application framework for R [5]. The advantage of using Shiny as web representation for LIFE data is that it combines the computational power of R with interactive features. Each Shiny application has two essential R scripts: `ui.R` and `server.R`. The script `ui.R` contains the code for the web presentation and `server.R` deals with back end data and output processing. One way Shiny implements interactivity is called reactive programming: a three step procedure consists of reactive value (input), reactive expression and output object. The input values from a web page are stored in reactive values (in `ui.R`). They further invoke the reactive expression via the reactive() function (in `server.R`). Both reactive values and reactive expression can be used in output object rendering (in `server.R`). The output objects are presented back out to the web page (in `ui.R`). The reactive() function ensures that whenever a reactive value is changed (through user input or by program), the re-
6.1. SOFTWARE COMPOSITION OF LIFE SHINY APPLICATION

execution of the reactive expression that directly or indirectly depends on that reactive value will also be triggered automatically.

Figure 6.1 shows a short example for reactive programming. On the data input page of the LSA user interface (screenshot see Figure 6.3), the users are allowed to choose between Leipzig map with Ortsteile or Stadtbezirke via input element radioButtons. This input is stored as the reactive value `input$whichmap` and is used to retrieve the right labeling data in the `server.R` script via Shiny reactive() function. The output `output$map` is a renderPlot object and generates the map of Leipzig using the retrieved reactive expression `label_data` as an input parameter for the `plot_lifemap()` function.

Figure 6.1: Example codes for reactive programming in Shiny. These are the actual codes used for the selection of labeling data for Ortsteil or Stadtbezirk on the data input page (Figure 6.3).

LSA is constructed based on this Shiny `ui.R ↔ server.R` framework. Figure 6.2 shows the software composition of LSA and its relationship to the architecture components introduced in Chapter 3. The application is composed of seven main R scripts, two `ui` scripts for the LSA user interface, two `server` scripts for the LSA server and three scripts containing functions for data preprocessing and data accessing. The visualization of statistics in LIFE focuses on two data types: continuous data and categorical data. The continuous data are visualized as choropleth style maps and the categorical data are visualized as pie charts or bar charts on maps. For these different ways of visualization, different data preprocessing methods and plotting functions are used for each data type (see Chapter 4 and Chapter 5 for details). This design concept of separation based on data types is also applied to the composition of LSA (Figure 6.2). Consequently, each data type possesses its own `ui`, `server` and data processing scripts (Figure 6.2). There is no communication between these two sets of scripts, i.e. for example, script `server_continuous.r` only interacts with `ui_continuous.r` but not with `ui_categorical.r`. For a clearer software structure, the standardization of continuous data is implemented as a separate data processing script `get_data_standardization.r` and it interacts with `server_continuous.r` directly.
6.2 The LSA data access unit and tab Data

The LSA data access unit of the functionalities layer (Figure 3.2) enables the direct database access to the data source layer of LIFE-SDVS. The LSA data access unit receives the data requests from the LSA server, obtains the data from the LIFE database and returns the data to the LSA server. When the users select a data source through the three data input options (see descriptions below), an SQL query is generated by functions defined in get_data_continuous.r and get_data_categorical.r scripts. The query is then entered into the data accessing function supplied by the package ROracle, an Oracle Database interface (DBI) driver for R provided by the ORACLE® Corporation [13], to fetch data from database. The data stays in get_data_*.r for preprocessing and the results are sent to...
Table 6.1: Seven main R scripts of LIFE Shiny Application and their major functions

<table>
<thead>
<tr>
<th>continuous data</th>
<th>categorical data</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>ui_continuous.r</strong></td>
<td><strong>ui_categorical.r</strong></td>
</tr>
<tr>
<td>• construct web page for continuous data</td>
<td>• construct web page for categorical data</td>
</tr>
<tr>
<td>• display choropleth map</td>
<td>• display pie or bar charts on map</td>
</tr>
<tr>
<td>• display data table of statistics</td>
<td>• display data table of statistics</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>server_continuous.r</th>
<th>server_categorical.r</th>
</tr>
</thead>
<tbody>
<tr>
<td>• reactive functions to retrieve reactive expressions based on input values selected in <strong>ui_continuous.r</strong></td>
<td>• reactive functions to retrieve reactive expressions based on input values selected in <strong>ui_categorical.r</strong></td>
</tr>
<tr>
<td>• utilize functions defined in</td>
<td>• utilize functions defined in</td>
</tr>
<tr>
<td><strong>get_data_continuous.r</strong></td>
<td><strong>get_data_categorical.r</strong></td>
</tr>
<tr>
<td>to generate data for output objects</td>
<td>to generate data for output objects</td>
</tr>
<tr>
<td>(e.g. maps and data tables)</td>
<td>(e.g. maps and data tables)</td>
</tr>
<tr>
<td>• utilize functions defined in</td>
<td>• use <strong>plot_categorical</strong> function in</td>
</tr>
<tr>
<td><strong>get_data_standardization.r</strong></td>
<td>package <strong>lifemap</strong> to produce reactive map output</td>
</tr>
<tr>
<td>to display standardized means on map</td>
<td></td>
</tr>
<tr>
<td>• use <strong>plot_continuous</strong> function in</td>
<td></td>
</tr>
<tr>
<td>package <strong>lifemap</strong> to produce reactive map output</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>get_data_continuous.r</th>
<th>get_data_categorical.r</th>
</tr>
</thead>
<tbody>
<tr>
<td>• obtain LIFE data directly from database using package <strong>ROracle</strong></td>
<td>• obtain LIFE data directly from database using package <strong>ROracle</strong></td>
</tr>
<tr>
<td>• generate SQL query for <strong>User-defined data input</strong></td>
<td>• generate SQL query for <strong>User-defined data input</strong></td>
</tr>
<tr>
<td>• define functions for continuous data preprocessing</td>
<td>• define functions for categorical data preprocessing</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>get_data_standardization.r</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>• define functions for age standardization on means</td>
<td></td>
</tr>
</tbody>
</table>

LSA server for visualization using the plotting functions.

On both continuous and categorical data user interfaces, there are three sources of inputing data for visualization: Example data, **User-defined** and **Upload data** (sidebar tab *Data* in Figure 6.3). Default is Example data. For users to explore the functionalities of the Shiny application, two example data sets are provided (option Example data): hand grip strength data for the continuous data interface and body mass index (BMI) data in categorical form for the categorical data interface. To use the **User-defined data** input option, the users only need to provide the following attributes to the user interface (in brackets are example entries of BMI data in continuous form):

- Derivattabelle Name (D00074)
- Feldname of the data to be plotted (D00074_F0004)
6.2. THE LSA DATA ACCESS UNIT AND TAB DATA

- Feldname of SIC (D00074_F0001)
- Feldname of Erhebungsdatum (D00074_F0002)

These attributes are defined in the Derivattabelle in the LIFE research database. The Derivattabelle of BMI data is attached in Appendix B. “SIC” stands for subject identifier codes and is generated by a special two-pass encryption program for all participants of the study in order to pseudo-anonymise their data. The attribute “Feldname of Erhebungsdatum” is used to extract the age of the participants in the SQL query. The User-defined option on the categorical data user interface uses the attributes of the Socioeconomic Status (SES) data set as an example.

Figure 6.3: Screenshot of the tab Data on continuous data user interface in LIFE Shiny Application. This image shows the data input options, varieties of statistics and two region levels (Ortsteil and Stadtbezirk) available for continuous data. The map shows the means of hand grip strength in each Leipzig Ortsteile and the numbers are Ortsteile IDs.

Since the users might wish to do extra data cleaning before the data is visualized, LSA also allows the users to upload their own csv files. The data input in the option Upload data shall be in compliance with the format introduced in the Table 4.1 for continuous data and that in the Table 4.2 for categorical data.

In addition to the three input data options on the tab Data of the continuous data user interface (Figure 6.3), users can also choose which statistics (mean, median or absolute
frequency) to be visualized on the choropleth map. For both continuous and categorial data
user interfaces, there are also options to which Leipzig map data shall be visualized: Ortsteil
or Stadtbezirk.

6.3 Functionalities on tab Filter

Figure 6.4 displays the screenshot of the tab Filter on the user interface for continuous data.
The tab Filter of categorial data is the same except there is no option for applying age
standardization which is only available for the means of continuous data. The users can
select to display the statistics for different genders and for three age groups: (18,40], (40,60],
(60,80+]. These selections are used as filters in the data preprocessing functions introduced
in Chapter 4. Setting the filter on the absolute frequency forces the regions with sample size
less than the threshold to be colored as too little data (for continuous data) or the bar charts
or pie charts are absent in these regions (for categorial data).

![Figure 6.4: Screenshot of the Filter options available in LIFE Shiny Application. The map shows the age standardized means of hand grip strength in each Ortsteil of Leipzig and the numbers are Ortsteile IDs. Regions in gray color are those with less than 30 individuals sampled.](image)

On the user interfaces of LSA, in addition to the map, a data table is also shown. Each row in the data table displays the information of a region on the map. The common content for both data types in the data table comprises the region IDs, region names and absolute frequency. Extra columns for continuous data are mean, standard deviation, standard errors,
median and 95% confidence intervals. For categorical data, a column containing categories of the data set is also shown. The content displayed in the data table also reacts to the selections on the tab *Filter* interactively. For example, if the gender filter is applied and only a subgroup of the gender is selected to display (i.e. only male or only female), only the corresponding selected data is shown in the data table. Similarly, the data of a specific age group is displayed (i.e. (18,40], (40,60] or (60,80+] if the respective age filter is applied.

### 6.4 Map functionalities for continuous data

Instead of inputing the values for the arguments in the `plot_continuous` function manually, users can easily change them via the elements on the tabs *Map Features* and *Label Placement* of the continuous data user interface. On the tab *Map Features*, users can choose if the map shall be filled with color or without (Figure 6.5). The option *User-defined* of the *Fill color* element takes names of ColorBrewer palettes [1] as value for the `pal` argument in the `plot_continuous` function (Figure 5.7 line 15). Users can also choose to use region id, name or region names combined with statistic values as label text.

![Figure 6.5: Tab Map Features for continuous data user interface in LIFE Shiny Application.](image)

Elements in the tab *Label Placement* enable the users to generate maps with boundary labeling easily. Assigning regions to the three labeling groups: right_group, left_group and bottom_group allows users to choose which labels to show as boundary labels (Figure 6.6).
6.5 Map functionalities for categorical data

The map shown on the categorical data user interface is generated by the `plot_categorical` function in the package `lifemap` (Figure 5.13). Instead of the map coloring options as on the continuous data user interface (Figure 6.5), the users can choose bar or pie charts to display the number of occurrences (absolute frequency) in each category for categorical data (Figure 6.7). In addition to the font and size of the legend text, the size of the pie charts or bar charts can be adjusted.

Figure 6.6: Screenshot of the tab **Label Placement** for continuous data user interface in LIFE Shiny Application. The map on the right shows the customized labeling groups with boundary labeling and internal labels are set as invisible. To save space, margin adjustment options are not shown.
can also be adjusted. Figure 6.7(a) shows the screenshot of the tab *Map Features* for bar chart and Figure 6.7(b) shows that for pie chart. The absolute frequency in each category of the socioeconomic status in each Stadtbezirk of Leipzig are displayed in both chart types. The pie charts provide an intuitive way to see if a certain category dominates among the regions on the map. On the other hand, bar charts display the trend within a region more distinctively.

![Map Features for bar chart](image1)

**Absolute frequency in each category of Socioeconomic status (gender: all, age group: all)**

(a) bar chart

![Map Features for pie chart](image2)

**Absolute frequency in each category of Socioeconomic status (gender: all, age group: all)**

(b) pie chart

**Figure 6.7:** The tab *Map Features* on the categorical data user interface of LIFE Shiny Application. Subfigure (a) for bar charts and subfigure (b) for pie charts. Both maps show the socioeconomic status in five categories in each Stadtbezirk of Leipzig.
7 | Use Cases and Analysis

In this chapter, two use cases are selected to demonstrate the spatial visualization of LIFE data in LIFE-SDVS: hand grip strength for continuous data type (Section 7.1) and body mass index for categorical data type (Section 7.2). Various features and different types of maps are presented to show the functionalities available in LIFE-SDVS. Section 7.1.1 aims to show the application of the filter functions, i.e. the visualization of hand grip strength data are displayed for different genders and different age groups. Section 7.1.2 demonstrates the age standardization on the means of hand grip strength for male and female probands in each Ortsteil of Leipzig. Boundary labeling and median are used in the examples in Section 7.1.1 and internal labeling and means are used in Section 7.1.2. Customized aesthetic settings, such as different filling colors for the maps, different fonts for the labels, or different leader colors are also applied. For the categorical data type, the proportions of each BMI category are plotted as pie charts in each of the Leipzig Ortsteile for both genders (Section 7.2).

7.1 Use case for continuous data: hand grip strength

Hand grip strength can be measured easily with a handgrip dynamometer and as stated by Martin-Ruiz et al. [92] "is probably the best characterized biomarker of mortality so far". Low grip strength in health adults across all ages predicts increased risk of premature mortality, disability in later life and risk of post-surgery complications or longer hospitalization ([42], [61], [92], [113], [117], [118]). Furthermore, the Prospective Urban-Rural Epidemiology (PURE) study, a large global study investigating nearly 140,000 adults in 17 countries, shows that hand grip strength can help predict not only the risk of death but also the risk of cardiovascular disease [82]. The researchers quantified the prognostic values and found that every 5-kg decrease in grip strength was linked to a 16% increase in all-cause mortality (i.e. hazard ratio = 1.16, 95% CI 1.13–1.20), a 17% increase in both cardiovascular (CI 1.11–1.24) and non-cardiovascular mortality (CI 1.12–1.21), a 9% increase in risk of stroke (CI 1.05–1.15) and a 7% increase in myocardial infarction (CI 1.02–1.11). All these associations are with statistical significance of p<0.0001. These figures show that grip strength was a more powerful predictor of cardiovascular mortality than of cardiovascular disease incidence. Another discovery in the study was that grip strength were positively associated with cancer in high-income countries (though with a small hazard ratio of 0.916, CI 0.880–0.953; p<0.0001), but not in middle-income and low-income countries. However, whether efforts to improve grip strength can
7.1. USE CASE FOR CONTINUOUS DATA: HAND GRIP STRENGTH

Table 7.1: Partial data output of the function `get_group_data_con` in the continuous data preprocessing. The data is generated from the hand grip strength containing male population of all age groups in each Ortsteil of Leipzig.

<table>
<thead>
<tr>
<th>region</th>
<th>gender</th>
<th>num</th>
<th>mean</th>
<th>sd</th>
<th>median</th>
<th>q1</th>
<th>q3</th>
</tr>
</thead>
<tbody>
<tr>
<td>00</td>
<td>1</td>
<td>10</td>
<td>41.15</td>
<td>6.90</td>
<td>40.05</td>
<td>35.83</td>
<td>44.00</td>
</tr>
<tr>
<td>01</td>
<td>1</td>
<td>31</td>
<td>43.58</td>
<td>11.24</td>
<td>40.00</td>
<td>35.50</td>
<td>52.35</td>
</tr>
<tr>
<td>02</td>
<td>1</td>
<td>62</td>
<td>42.15</td>
<td>9.76</td>
<td>43.30</td>
<td>34.98</td>
<td>47.77</td>
</tr>
<tr>
<td>03</td>
<td>1</td>
<td>88</td>
<td>43.81</td>
<td>10.57</td>
<td>44.00</td>
<td>36.00</td>
<td>50.83</td>
</tr>
<tr>
<td>04</td>
<td>1</td>
<td>64</td>
<td>45.80</td>
<td>11.33</td>
<td>45.00</td>
<td>38.30</td>
<td>52.50</td>
</tr>
<tr>
<td>05</td>
<td>1</td>
<td>66</td>
<td>46.42</td>
<td>7.32</td>
<td>46.20</td>
<td>42.23</td>
<td>51.25</td>
</tr>
</tbody>
</table>

reduce an individual’s risk of death and cardiovascular disease needs to be further studied.

7.1.1 Visualization of hand grip strength data

To demonstrate the visualization functionalities of LIFE-SDVS for continuous data, the hand grip strength data collected in the LIFE-Adult-Study is used. This section presents various maps downloaded from LIFE Shiny Application after the application of various filter functions. The filter function in LIFE-SDVS is available for gender and age. The raw data of hand grip strength is obtained directly from the LIFE database with the LSA data access unit (see Section 6.2 for the mechanism). Each row in the raw data is the information of one participant and the information collected includes the region, gender, age and the measured hand grip value of the participant (see Table 4.1 for an example). This raw data is then preprocessed using function `get_group_data_con()` (Page 26). Table 7.1 shows the partial output of the hand grip strength data with gender filter set as male and of all age groups.

Figure 7.1 shows the hand grip strength median of LIFE participants in different gender groups on the map of Leipzig Ortsteile. Regions with absolute frequency less than 30 individuals are colored in gray. Figure 7.1(a) presents the grip strength median of both genders, Figure 7.1(b) and Figure 7.1(c) present that of male and female participants, respectively. Boundary labels are assigned to the Ortsteile with median values less than or equal to 32 kg and 42 kg in the corresponding subfigures (a), (b) and less than 27 kg in subfigure (c). As seen from the figures, the grip strength of male and that of female differ significantly, with male having values between 38 and 48 kg (Figure 7.1(b)) and females having values between 25 and 31 kg (Figure 7.1(c)). Therefore, the medians of both genders pulled together are not representative (Figure 7.1(a)).

Four Ortsteile in the west of Leipzig, i.e. Grünau-Ost, Grünau-Nord, Grünau-Mitte and Großzschocher, belong to the 15 regions with lowest hand grip strength medians both of male (Figure 7.1(d)) and of female population (Figure 7.1(c)). On the east part of Leipzig, another cluster of low grip strength regions for both genders consists of the three Ortsteile: Schönefeld-Ost, Mölkau and Sellerhausen-Stünz. These are also the regions with higher average ages of probands. The average ages of the probands in these seven regions are
all more than 60 years while the average age of all probands in the LIFE hand grip strength data set is ca. 57.6 years. The comparison to the data of the city government in the Leipzig Ortsteilkatalog [125] also shows that the average ages in these regions are mostly higher than those of other Leipzig regions. The average inhabitant ages in most of these regions are ca. 50 years while the average age of whole Leipzig population is 43.4 years. Consequently, the lower hand grip strength in these regions as shown on the map of Leipzig (Figure 7.1) can be due to the fact that the population there are older.

Dodds et al. in 2014 published the first study that provides normative data for grip strength across the life course based on 12 studies conducted in Great Britain [48]. In their publication two years later [49], the authors suggest that the results of British studies are applicable in other developed countries, such as Canada, United States, Australia and most of the west european countries including Germany. The hand grip strength values are significantly lower in developing regions such as China, India and Cuba. According to [48], the medians of males of age 40-60 are between 44 and 40 kg (decreasing with age) and those of age 60-80 are between 45 and 32 kg. The medians of females of age 40-60 are between 31 and 28 kg and those of age 60-80 are between 27 and 19 kg. The means have similar values as the medians. The minimum hand grip strength median of male participants is in Grünau-Nord with value 38 kg (Figure 7.1(b)) and that of female participants is in Anger-Crottendorf with value 25 kg (Figure 7.1(c)). To conclude, the hand grip strength medians of the Leipzig probands are still within normative standard even in the regions where minimum values are measured.

Figure 7.2 shows the hand grip strength medians of female participants of different age groups in Leipzig Ortsteile. Regions with absolute frequency less than 30 individuals are colored in gray. Since there is only one region that has sufficient sample size in age group (18,40], the map of this age group is not shown. Boundary labels are assigned to the Ortsteile with the median value less than or equal to 29 kg and 24 kg in the subfigures (a) age group (40,60] and (b) age group (60,80+], respectively. The grip strength medians of female probands of age group (40,60] are between 31 and 26 kg (Figure 7.2(a)) and that of age group (60,80+] are between 28 and 23 (Figure 7.2(b)). Compared to the results of [48] mentioned in the previous paragraph, the values of Leipzig female probands of age group (40,60] are similar to that of British females (31-28 kg). However, the values of Leipzig female probands of age group (60,80+] are a little higher than that of British females (27-19 kg). Nevertheless, cautions shall be taken that the higher value of Leipzig females might be due to the fact that there are more participants with age of 60-75 years than with age of 75-80 years in the study cohort (see Figure 2.1).
7.1. USE CASE FOR CONTINUOUS DATA: HAND GRIP STRENGTH

Figure 7.1: Hand grip strength medians of probands of different gender groups in Leipzig Ortsteile. Subfigure (a) both genders (b) male (c) female.
7.1. USE CASE FOR CONTINUOUS DATA: HAND GRIP STRENGTH

Figure 7.2: Hand grip strength medians of female probands of different age groups in Leipzig Ortsteile. Subfigure (a) age group (40,60] (b) age group (60,80+).

7.1.2 Age-standardization of hand grip strength data

Age-standardization in LIFE-SDVS can be applied in all three gender groups: both_gender, male and female. The means of hand grip strength data in all three gender groups are used to demonstrate the application of age standardization. Table 7.2 shows the means of hand grip strength of three age groups in each gender category from LIFE data in two Leipzig Ortsteile Schleußig and Eutritzsch. For example, the standardized mean of hand grip strength in Schleußig for both_gender is computed by multiplying the weights in column both_gender in Table 4.4 with the means in column both_gender in Table 7.2 (highlighted in green color).
Table 7.2: Means of hand grip strength in each age group for three gender categories in Leipzig Schleußig and Eutritzsch. The means with and without standardization of each gender category in each Ortsteil are also shown. Data obtained from LIFE-Adult-Study.

<table>
<thead>
<tr>
<th>age group</th>
<th>Schleußig (OT50)</th>
<th>Eutritzsch (OT93)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>both_gender</td>
<td>male</td>
</tr>
<tr>
<td>(18,40]</td>
<td>38.33</td>
<td>49.51</td>
</tr>
<tr>
<td>(40,60]</td>
<td>41.06</td>
<td>49.90</td>
</tr>
<tr>
<td>(60,80+]</td>
<td>32.69</td>
<td>42.16</td>
</tr>
<tr>
<td>w/o standardized mean</td>
<td>38.52</td>
<td>47.89</td>
</tr>
<tr>
<td>standardized mean</td>
<td>37.40</td>
<td>47.58</td>
</tr>
</tbody>
</table>

The calculation is the following:

\[
\text{standardized mean of hand grip strength of Schleüßig population} = 38.33 \cdot 0.39 + 41.06 \cdot 0.30 + 32.69 \cdot 0.31 = 37.40
\]

The second example is age standardization applied only on female population in Eutritzsch. Likewise, the proportions in the female column in each age group in Table 7.2 are multiplied with the weights in the column female in Table 4.4 (highlighted in purple color in both tables). The sum of the products of each row is the standardized mean:

\[
\text{standardized mean of hand grip strength of Eutritzsch female population} = 30.10 \cdot 0.37 + 30.22 \cdot 0.28 + 25.37 \cdot 0.35 = 28.48
\]

Figure 7.3 presents the maps colored according to the means of the hand grip strength, with and without age standardization of both genders in the Leipzig Ortsteile. There are more regions colored as too little data after standardization than before standardization. This is due to the fact that in addition to setting a filter of minimum 30 participants in each Ortsteil, as long as one of the age groups in a region contains no individual, the age standardization is not applicable. Hence, these regions are colored as too little data. Figure 7.3(a) shows that the grip strength means of males in all Ortsteile are less than 48 kg before standardization. However, the maximum mean value among all Ortsteile is higher after standardization (ca. 54 kg, Figure 7.3(b)). Similar trend is observed in the female population. While the maximum mean value among all Ortsteile before standardization is ca. 30 kg (Figure 7.3(c)), that after standardization is ca. 32 kg (Figure 7.3(d)). In many regions, the age standardized means are also higher than those before standardization. The trend could be due to the underrepresentation of the age group (18,40] in the LIFE samples (see Figure 2.1). This also shows the necessity of age-standardization.
People who are obese have higher risk of many diseases and health problems such as higher all-cause mortality, high blood pressure, sleep apnea and breathing problems, some cancers (e.g. breast, colon, kidney, gallbladder, and liver), type 2 diabetes, mental illness (e.g. clinical depression, anxiety, and other mental disorders), and stroke ([32], [77], [87], [98]). In addition, studies in many countries have shown the economic burden of obesity, e.g. France [83], the

Figure 7.3: Maps of Leipzig showing the means of hand grip strength of both genders in each Ortsteile. (a) the means without age-standardized (b) the means with age-standardized. The numbers within each region are Ortsteile IDs.

7.2 Use case for categorical data: body mass index

The BMI data in categorical form in LIFE database is used to demonstrate the functionalities of LIFE-SDVS for the categorical data type. BMI is computed by an individual’s weight in kilograms divided by the square of height in meters (i.e. in units $kg/m^2$). BMI can be an indicator of high body fatness and hence can be used for assessment of overweight and obesity [62]. The World Health Organization classifies the BMI values into four weight status categories for adults (shown in Table 7.3) [8]. This classification is applicable to both genders and is age independent. This classification is also used in the pie charts in this section. People who are obese have higher risk of many diseases and health problems such as higher all-cause mortality, high blood pressure, sleep apnea and breathing problems, some cancers (e.g. breast, colon, kidney, gallbladder, and liver), type 2 diabetes, mental illness (e.g. clinical depression, anxiety, and other mental disorders), and stroke ([32], [77], [87], [98]). In addition, studies in many countries have shown the economic burden of obesity, e.g. France [83], the
7.2. USE CASE FOR CATEGORICAL DATA: BODY MASS INDEX

Netherlands [121], New Zealand [127], the USA ([41], [53]) and Germany ([115], [140]).

Table 7.4 shows the percentage of each BMI category of male and female population in age groups (40,60] and (60,80+] in Germany in 2013 and that of Leipzig from LIFE. Overall, there is only small proportion of the population underweight. The proportions of each BMI category in the male population of age group (40,60] are similar in Germany and Leipzig. There are about 50% of the male population of age group (60,80+] are in category "overweight", both in Germany and in Leipzig. However, in this group, the proportion of obesity is ca. 8 % higher while the proportion of normal weight is ca. 8% less in Leipzig than those in the German population. Among German females of age group (40,60], 55.6% are with BMI values in the normal range, but only 45.6% of the females of same age group in Leipzig are within BMI normal range. On the other hand, the proportion of overweight and obese are 3.9% and 7.5%, respectively, higher in the females of age group (40,60] in Leipzig than those in Germany. The trend of having higher BMI values in the Leipzig population is also observed in the females of age group (60,80+]. In this group, instead of 40.7% has BMI values in the normal range in Germany, this proportion is only 27.6% in Leipzig. Furthermore, the proportion of overweight and obese are 2.6% and 11.9% higher in Leipzig than those in Germany. Overall, there are higher proportions of overweight and obesity among the population of age over 40 in Leipzig than the German average.

Table 7.3: BMI and standard weight status categories classified by World Health Organization [8].

<table>
<thead>
<tr>
<th>BMI</th>
<th>Status</th>
</tr>
</thead>
<tbody>
<tr>
<td>≤ 18.5</td>
<td>Underweight</td>
</tr>
<tr>
<td>18.5 – 24.9</td>
<td>Normal range</td>
</tr>
<tr>
<td>25.0 – 29.9</td>
<td>Overweight</td>
</tr>
<tr>
<td>≥ 30.0</td>
<td>Obese</td>
</tr>
</tbody>
</table>

To examine if the situation of having higher proportion of overweight and obesity than German average is prevalent among Leipzig Ortsteile or only few Ortsteile have such property, the BMI data in LIFE is visualized using LIFE-SDVS. The absolute frequencies of probands in each BMI category are aggregated for each region of Leipzig in order to plot the pie charts. Table 7.5 shows an example of aggregated results containing the male population of age group (40,60] in Ortsteile Schleußig and Plagwitz in Leipzig.

Figure 7.4 shows the pie charts depicting absolute frequencies of each BMI category of males in age groups (40,60] and (60,80+] in each Leipzig Ortsteil. The Ortsteile with absolute frequencies less than 30 are filtered out and not plotted. The corresponding proportions based on German population with data derived from Gesundheitsberichterstattung des Bundes (GBE) [6] are presented on top of each subgraph for comparison. The data downloaded from GBE are already aggregated in four BMI categories and only needed to be pulled into corresponding age groups for the comparison here. If the sum of the proportions in the categories "overweight" and "obese" of the respective gender and age group of an Ortsteile is higher than that of German data, the Ortsteil is colored in green. To simplify the description in
7.2. USE CASE FOR CATEGORICAL DATA: BODY MASS INDEX

Table 7.4: Percentage of population in each BMI category for two age groups in Germany and in Leipzig. The German data is derived from Gesundheitsberichterstattung des Bundes in 2013 [6]. The Leipzig data is from LIFE-Adult-Study.

<table>
<thead>
<tr>
<th>BMI</th>
<th>Germany Male</th>
<th>Germany Female</th>
<th>Leipzig Male</th>
<th>Leipzig Female</th>
</tr>
</thead>
<tbody>
<tr>
<td>(40,60]</td>
<td>0.4</td>
<td>0.4</td>
<td>0.5</td>
<td>1.0</td>
</tr>
<tr>
<td>(60,80+]</td>
<td>1.8</td>
<td>0.1</td>
<td>2.4</td>
<td>0.4</td>
</tr>
<tr>
<td>(40,60]</td>
<td>2.4</td>
<td>1.8</td>
<td>28.0</td>
<td>31.9</td>
</tr>
<tr>
<td>(60,80+]</td>
<td>45.7</td>
<td>27.6</td>
<td>30.1</td>
<td>40.5</td>
</tr>
<tr>
<td>(40,60]</td>
<td>19.3</td>
<td>14.0</td>
<td>40.7</td>
<td>31.9</td>
</tr>
<tr>
<td>(60,80+]</td>
<td>14.0</td>
<td>45.7</td>
<td>49.8</td>
<td>27.6</td>
</tr>
</tbody>
</table>

Table 7.5: Example of aggregated data as input for the visualization of categorical data type in LIFE-SDVS. The table shows the absolute frequency in each BMI category of the male population of age group (40,60] in Ortsteile Schleußig and Plagwitz.

<table>
<thead>
<tr>
<th>Ortsteil ID</th>
<th>Name</th>
<th>Category</th>
<th>Absolute frequency</th>
</tr>
</thead>
<tbody>
<tr>
<td>50</td>
<td>Schleußig</td>
<td>2</td>
<td>21</td>
</tr>
<tr>
<td>50</td>
<td>Schleußig</td>
<td>3</td>
<td>29</td>
</tr>
<tr>
<td>50</td>
<td>Schleußig</td>
<td>4</td>
<td>10</td>
</tr>
<tr>
<td>51</td>
<td>Plagwitz</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>51</td>
<td>Plagwitz</td>
<td>2</td>
<td>19</td>
</tr>
<tr>
<td>51</td>
<td>Plagwitz</td>
<td>3</td>
<td>22</td>
</tr>
<tr>
<td>51</td>
<td>Plagwitz</td>
<td>4</td>
<td>11</td>
</tr>
</tbody>
</table>

the following text, this condition is denoted as \( LhG \): Leipzig samples heavier than the overall German population. Among the 36 Ortsteile with absolute frequencies over 30 individuals, 55.6\% are having \( LhG \) in the age group (40,60] (Figure 7.4(a)). For the age group (60,80+] (Figure 7.4(b)), 35 Ortsteile have more than 30 participants and 89\% of these Ortsteile are with \( LhG \).

Figure 7.5 visualizes the same statistics for the female population in Leipzig and also with German data above each subfigure. In the female population, the situation of \( LhG \) is even more prevalent. For the age group (40,60], 91\% of the 43 Ortsteile with minimum 30 participants are with \( LhG \) (Figure 7.5(a)). Among the 35 Ortsteile with absolute frequencies over 30 individuals, even 97\% are with \( LhG \), i.e. only one Ortsteil, Leutzsch, is an exception (Figure 7.5(b)). To conclude, compared to German countrywide data, the case that having higher proportion of "overweight" and "obese" in Leipzig adult population of different age groups is prevailing among the Ortsteile (Figure 7.4 and Figure 7.5).
7.2. USE CASE FOR CATEGORICAL DATA: BODY MASS INDEX

Figure 7.4: Proportions of male participants in each BMI category of different age groups in Leipzig Ortsteile. Subfigure (a) age group (40,60] (b) (60,80+]. Above each subfigure is the corresponding proportions based on German population with data derived from Gesundheitsberichterstattung des Bundes in 2013 [6]. If the sum of the proportions in the categories "overweight" and "obese" in an Ortsteil is higher than that of Germany, the Ortsteil is colored in green.
Figure 7.5: Proportions of female participants in each BMI category of different age groups in Leipzig Ortsteile. Subfigure (a) age group (40,60] (b) (60,80+]. Above each subfigure is the corresponding proportions based on German population with data derived from Gesundheitsberichterstattung des Bundes in 2013 [6]. If the sum of the proportions in categories "overweight" and "obese" in an Ortsteil is higher than that of Germany, the Ortsteil is colored in green.
Due to the large amount and variety of data obtained in the LIFE-Studies (Section 2.1 and Section 2.2), there was a necessity to build an automated spatial visualization tool. Thus, in this master thesis the software system LIFE-SDVS has been conceptualized and implemented. LIFE-SDVS automizes the visualization of LIFE assessment results on the map of Leipzig by fetching the data directly from the LIFE database. LIFE-SDVS provides the users an interactive web interface to explore statistics of LIFE data on Leipzig map and to use LIFE-SDVS as a map generation tool.

Two major data types in the LIFE database were identified: continuous data type and categorical data type. Different visualization approaches were developed for each of these data types. For continuous data, three different statistical attributes, i.e. mean, median and absolute frequency, are displayed in choropleth map style using the plotting function `plot_continuous` (Figure 5.7). For categorical data, the absolute frequencies of each category as pie charts or bar charts are shown in regions of Leipzig using the plotting function `plot_categorical` (Figure 5.13). To support the different types of maps, different data preprocessing approaches have been constructed (Chapter 4). While comparing statistics between populations of various regions, the underlying cofounder structure has to be considered. Hence, LIFE-SDVS also provides the age-standardization option for the means of continuous data (Section 2.3, Section 4.3 and Section 7.1.2).

The package `lifemap` empowers the map visualization tasks of LIFE-SDVS (Chapter 5). The design as an independent package eases the maintenance of the software, gives the architecture a clear structure and can be simply imported into the LIFE Shiny Application (LSA) as a visualization extension (Chapter 3). In addition to turning spatial data into maps and plotting statistical graphics within regions, the stand-alone package `lifemap` is also capable of finding good labeling positions and producing boundary labeling. Good labeling positions within regions of a map can be used for placing internal labels or graphics and also provide good starting points for the leaders of boundary labels. The Label Positioning Algorithm (LPA) is proposed to locate such good labeling positions and the results on the maps of Leipzig are good (Section 5.1). The search for such positions is not covered by most of the literatures in map labeling research areas except one paper: that by Dörschlag et al. [50]. However, the vague description of the paper makes implementation difficult. Moreover, the concepts behind LPA are straightforward while those of by Dörschlag et al. consisted of three different stages, that, could arguably influence the efficiency of the algorithm.
One of the aims of this thesis was to enable boundary labeling for placing long label text. The visualization package lifemap is, to the extent of the author’s knowledge, the first R package to achieve this aim for maps. In the package, the plotting function plot_continuous (Figure 5.7) empowers boundary labeling and the embedded Label Alignment Algorithm (Section 5.2.2) in the plotting function eliminates the leader intersection problem. Further improvement of the boundary labeling feature is the use of LSA user interface to adjust the parameters for the arguments in plot_continuous and make the application of boundary labeling an interactive experience. The user interface, backed by the LSA server, has strongly reduced the effort of manually inputting long coordinate values to adjust the positions of boundary labeling groups or to change the margin sizes. Therefore, the researchers can produce maps much more efficiently. Moreover, the customizable boundary labeling assignment allows the researchers to decide which labels are shown in which labeling group and therefore determine what message shall be delivered. In the use case for continuous data type (Figure 7.1 and 7.2), the boundary labels were assigned to those regions where hand grip strength medians are less than a certain threshold. Hence the focus of the maps is placed on where the people with lower grip strength are in Leipzig.

LIFE Shiny Application provides all other functionalities of LIFE-SDVS except the visualization tasks built in package lifemap, i.e. direct data access, data preprocessing and an interactive web application (Figure 3.2). The software composition of the application is designed based on the Shiny ui.R ↔ server.R framework and the division between continuous and categorical data (Figure 6.2). Through the LSA user interface, the users can specify what data to be visualized by entering merely four attributes (page 49). With this information, the LSA data access unit generates an SQL query and enables the direct database access to fetch the data at run time (Section 6.2). The LSA data preprocessing unit (Figure 3.2) transforms then the raw data into the data format needed for map visualization. Filters on gender or age allow the users to specify the partial data they are interested in (Figure 6.4). Statistics such as mean, median and aggregated absolute frequencies can also be computed and visualized. Age-standardization of means is also one of the tasks of the LSA data preprocessing unit. Various interactive map features on the LSA user interfaces (Section 6.4 for continuous data and Section 6.5 for categorical data) allow the users to explore LIFE data and to generate maps efficiently.

The visualization of hand grip strength medians in different Ortsteile of Leipzig shows that even the regions with the smallest values still meet the normative standard. Two clusters of lower hand grip strength medians are identified on the west and east of Leipzig where the average ages of the populations in these regions are higher than the average of the whole Leipzig population. The comparison of maps of hand grip strength means before and after age-standardization indicates an underestimation of hand grip strength means before age-standardization. This shows the need for the application of standardization methods. The maps showing the visualization of categorical BMI data demonstrate that most of the Leipzig Ortsteile have higher proportions of overweight and obese people than those of Germany.
8.1 Future work on LIFE-SDVS

LIFE-SDVS sets a foundation on the spatial data visualization in LIFE. This thesis has proposed its software architecture, built up a backbone consisting an R package and a Shiny web application and the software components are implemented. Nevertheless, the current version of LIFE-SDVS is still at the very beginning phase for a more sophisticated, automized visualization system. Hence, some suggestions for the development of the future versions are given in the section.

The customizable boundary labeling allows the users to assign intended labels to show. In addition to manual assignment, some rule-based assignment functions can be implemented to define such a set of intended labels \( L \). An example of such a rule might be "assign boundary labels to the regions where the mean values larger than \( n \)", where \( n \) is given by the user. Second additional function on boundary labeling can be that the system assigns the elements in \( L \) automatically to the three labeling groups under certain criteria, such as no intersection allowed and the total leader length is minimized. The application of the methods proposed in [30] can be considered. Furthermore, the adjustments on the margins and the positions of labeling groups can also be fully automized.

In addition to plotting pie charts or bar charts in the regions of Leipzig, the boundary labeling feature can also be added to the categorical data maps. Moreover, rule-based coloring of the regions or even using the choropleth map of continuous data as background for the pie charts or bar charts can also be considered.

Though LPA has delivered a good result of locating good labeling positions on map of Leipzig, it has still limitations if the algorithm shall be applied in more generic circumstances. The improvement suggestions were discussed in page 38.

In addition, LIFE-SDVS can be equipped with a spatial aggregation function in the future. From statistical point of view, it is sometimes desirable that several regions in a study area to be merged into one region for certain objectives. A simple example is that regions can be merged to reach a given sample size threshold. The topic is related to zone design problems ([21], [91], [101], [111]) and districting problem ([23], [76]). Table 2 in the paper of Cockings and Martin [40] lists various research aims of such problems for environmental and health studies. After the aims are determined, two further steps are followed: determining which regions to be merged together and merge execution. Merge execution in R can be done by utilizing the \texttt{unionSpatialPolygons} function from package \texttt{maptools}. To determine which regions to be merged together can be seen as a multiobjective optimization problem. A wide variety of spatial aggregation approaches have been proposed to tackle such problems ([65], [103], [111]). Some metaheuristics have also been applied such as simulated annealing ([18], [45], [88]), tabu search ([35], [103], [136]) and genetic algorithm ([17], [46], [129], [142], [143]).
The assessment programmes in LIFE-Adult-Study. The tables are adapted from [86].

### Table A.1 Physical and medical examinations

<table>
<thead>
<tr>
<th>Examination</th>
<th>Sample size</th>
</tr>
</thead>
<tbody>
<tr>
<td>Bio-specimen assevation</td>
<td>&gt;9000</td>
</tr>
<tr>
<td>Serum, plasma, lymphocytes for DNA, RNA, urine</td>
<td>&gt;9000</td>
</tr>
<tr>
<td>Anthropometry</td>
<td>&gt;9000</td>
</tr>
<tr>
<td>Classical: Body weight, body height, circumference measures</td>
<td>&gt;9800</td>
</tr>
<tr>
<td>3D laser based optical body surface scan</td>
<td>&gt;9800</td>
</tr>
<tr>
<td>MRI based abdominal fat tissue volumetry</td>
<td>&gt;1000</td>
</tr>
<tr>
<td>Cardiovascular system</td>
<td></td>
</tr>
<tr>
<td>Blood pressure</td>
<td>&gt;9900</td>
</tr>
<tr>
<td>Electrocardiography (10 s, 12 leads)</td>
<td>&gt;9700</td>
</tr>
<tr>
<td>Echocardiography</td>
<td>&gt;8300</td>
</tr>
<tr>
<td>Carotid ultrasound</td>
<td>&gt;9600</td>
</tr>
<tr>
<td>Ankle-brachial index (ABI), pulse wave velocity (PiW)</td>
<td>&gt;9200</td>
</tr>
<tr>
<td>Diabetes</td>
<td></td>
</tr>
<tr>
<td>Oral glucose tolerance test (OGTT)</td>
<td>&gt;2900</td>
</tr>
<tr>
<td>Physical activity and fitness</td>
<td></td>
</tr>
<tr>
<td>Hand grip strength</td>
<td>&gt;9700</td>
</tr>
<tr>
<td>7 day actimetry</td>
<td>&gt;2800</td>
</tr>
<tr>
<td>Eye</td>
<td></td>
</tr>
<tr>
<td>Optical coherence tomography and fundus photography</td>
<td>&gt;9300</td>
</tr>
<tr>
<td>Brain</td>
<td></td>
</tr>
<tr>
<td>MRI</td>
<td>&gt;2400</td>
</tr>
<tr>
<td>Electroencephalography (≥60 years only)</td>
<td>&gt;3000</td>
</tr>
<tr>
<td>Miscellaneous</td>
<td></td>
</tr>
<tr>
<td>Voice profile</td>
<td>&gt;2700</td>
</tr>
<tr>
<td>Olfactory test (Sniffin’ sticks 12)</td>
<td>&gt;7300</td>
</tr>
<tr>
<td>Skin prick test (six inhalative allergens)</td>
<td>&gt;8000</td>
</tr>
</tbody>
</table>

### Table A.2 Laboratory analyses

<table>
<thead>
<tr>
<th>Parameter</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Electrolytes: Sodium, Potassium, Chloride, Magnesium</td>
<td></td>
</tr>
<tr>
<td>Liver and pancreas: Alanine transaminase, Aspartate transaminase,</td>
<td></td>
</tr>
<tr>
<td>Choline esterase, Gamma-glutamyltransferase, Bilirubin (total and direct),</td>
<td></td>
</tr>
<tr>
<td>Lipase, Total protein, Albumin, Urea</td>
<td></td>
</tr>
<tr>
<td>Kidney: Creatinine, Cystatin C</td>
<td></td>
</tr>
<tr>
<td>Cardiac markers: Creatine kinase, Creatine kinase MB, Myoglobin,</td>
<td></td>
</tr>
<tr>
<td>Troponin T high sensitive, N-terminal prohormone of brain</td>
<td></td>
</tr>
<tr>
<td>natriuretic peptide</td>
<td></td>
</tr>
<tr>
<td>Lipid metabolism: Cholesterol, High density lipoprotein cholesterol,</td>
<td></td>
</tr>
<tr>
<td>Low density lipoprotein cholesterol, Apolipoprotein B, Apolipoprotein A1,</td>
<td></td>
</tr>
<tr>
<td>Triglycerides, Lipoprotein (a)</td>
<td></td>
</tr>
<tr>
<td>Glucose metabolism: Glucose, Insulin, C-peptide, Glycated</td>
<td></td>
</tr>
<tr>
<td>hemoglobin (HbA1c)</td>
<td></td>
</tr>
<tr>
<td>Iron metabolism: Transferin, Ferritin</td>
<td></td>
</tr>
<tr>
<td>Vitamins: Folic acid, Vitamin B12</td>
<td></td>
</tr>
<tr>
<td>Bone metabolism: Alkaline phosphatase, Phosphate, Calcium,</td>
<td></td>
</tr>
<tr>
<td>Osteocalcin, Beta-CrossLaps, Propeptide of type I collagen,</td>
<td></td>
</tr>
<tr>
<td>Parathormone, 25-Hydroxy vitamin D3</td>
<td></td>
</tr>
<tr>
<td>Endocrine function / hormones: Cortisol, Luteinizing hormone, Follicle</td>
<td></td>
</tr>
<tr>
<td>stimulating hormone, Estradiol, Testosterone, Sex hormone-binding</td>
<td></td>
</tr>
<tr>
<td>globulin, Dehydroepiandrosterone sulfate</td>
<td></td>
</tr>
<tr>
<td>Thyroid: Thyrotropin (TSH), Free triiodothyronine, Free thyroxine, TSH</td>
<td></td>
</tr>
<tr>
<td>receptor antibodies, thyroglobulin antibodies, Thyreoperoxidase antibodies</td>
<td></td>
</tr>
<tr>
<td>Inflammatory mediators: Interleukin 6, C-reactive protein high sensitive</td>
<td></td>
</tr>
<tr>
<td>Allergy diagnostics: specific immunoglobulin E sx1 (timothy grass,</td>
<td></td>
</tr>
<tr>
<td>rye, birch, mugwort, C. herbarum, D. pteronyssinus, cat, dog), fx5</td>
<td></td>
</tr>
<tr>
<td>(hen’s egg, cow’s milk, fish, wheat, peanut, soy), total immunoglobulin E</td>
<td></td>
</tr>
<tr>
<td>Hematology: Complete blood cell count with differential, Reticulocytes</td>
<td></td>
</tr>
<tr>
<td>Urine: Albumin, Creatinine, Immunoglobulin G</td>
<td></td>
</tr>
</tbody>
</table>
Table A.3 Computer-assisted personal interviews (I), self-administered questionnaires (Q) and cognitive tests (T) (Continued)

<table>
<thead>
<tr>
<th>Assessment</th>
<th>Type</th>
<th>Age range</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cognition</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Stroop test</td>
<td>T</td>
<td>18-79</td>
</tr>
<tr>
<td>Trail-Making Test A&amp;B</td>
<td>T</td>
<td>18-79</td>
</tr>
<tr>
<td>Subjective Memory Impairment</td>
<td>I</td>
<td>18-79</td>
</tr>
<tr>
<td>Verbal Fluency Test ‘Animals’</td>
<td>T</td>
<td>18-79</td>
</tr>
<tr>
<td>Behavioral Assessment of the dysexecutive syndrome (DEX)</td>
<td>Q</td>
<td>18-79</td>
</tr>
<tr>
<td>Triangle test</td>
<td>T</td>
<td>60-79</td>
</tr>
<tr>
<td>Reading the mind in the eyes test</td>
<td>T</td>
<td>60-79</td>
</tr>
<tr>
<td>Structured Interview for Diagnosis of Dementia of Alzheimer type, Multi-infarct Dementia and Dementia of other Aetiology according to DSM-III-R, DSM-IV and ICD-10</td>
<td>I/T</td>
<td>60-79</td>
</tr>
<tr>
<td>Barthel-Index for basic Activities of Daily Living</td>
<td>I</td>
<td>60-79</td>
</tr>
<tr>
<td>Instrumental Activities of Daily Living Scales (IADL)</td>
<td>I</td>
<td>60-79</td>
</tr>
<tr>
<td>CERADplus neuropsychological test battery</td>
<td>T</td>
<td>60-79</td>
</tr>
<tr>
<td>Wechsler Memory Scale</td>
<td>T</td>
<td>60-79</td>
</tr>
<tr>
<td>Iowa gambling task</td>
<td>T</td>
<td>18-79</td>
</tr>
<tr>
<td>n-back task</td>
<td>T</td>
<td>18-79</td>
</tr>
<tr>
<td>Reversal learning task</td>
<td>T</td>
<td>18-79</td>
</tr>
<tr>
<td>Depression</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Centre of Epidemiologic Studies - Depression Scale (CES-D)</td>
<td>Q</td>
<td>18-79</td>
</tr>
<tr>
<td>Childhood Trauma Questionnaire (CTQ)</td>
<td>Q</td>
<td>18-79</td>
</tr>
<tr>
<td>CIDI-DIA-X Screener</td>
<td>Q</td>
<td>18-79</td>
</tr>
<tr>
<td>Structured Clinical Interview for DSM Disorders (SCID)</td>
<td>I</td>
<td>60-79</td>
</tr>
<tr>
<td>Geriatric Depression Scale (15-item version, GDS-15)</td>
<td>Q</td>
<td>60-79</td>
</tr>
<tr>
<td>Inventory of Complicated Grief (ICG)</td>
<td>Q</td>
<td>60-79</td>
</tr>
<tr>
<td>Inventory of Depressive Symptoms (IDS-SR)</td>
<td>Q</td>
<td>60-79</td>
</tr>
<tr>
<td>Leipzig Life Event List</td>
<td>Q</td>
<td>60-79</td>
</tr>
</tbody>
</table>

*Only for the subgroup with additional obesity assessments*
An example of *Derivattabelle* in the LIFE research database. The attributes needed for the User-defined data input option in LIFE Shiny Application are obtained from such tables.
Body Mass Index

Annotated CRF

**Derivattabelle:** BMI (D00074)

**DQP-ID:** DQP_21_0051_20140325

**Beschreibung:** An Teilnehmern der LIFE-ADULT-Studie (Altersbereich 18 bis 79 Jahre) wurden Körpergröße, Körpergewicht und verschiedene Körperumfänge gemessen. Anhand dieser Merkmale wurden Maße für die Körperfettmasse insgesamt (Body Mass Index, BMI) und für die Körperfettverteilung (Taillenumfang, Taillenumfang/Hüftumfang) berechnet.

<table>
<thead>
<tr>
<th>Code</th>
<th>Beschreibung</th>
<th>Datentyp</th>
<th>Feldname</th>
</tr>
</thead>
<tbody>
<tr>
<td>S010061_SIC</td>
<td>Pseudonym des Teilnehmers</td>
<td>TEXT</td>
<td>BMI_S010061_SIC</td>
</tr>
<tr>
<td>S010061_DATUM</td>
<td>Datum, zu dem die Daten laut Laufzettel erhoben wurden (Erhebungsdatum)</td>
<td>DATUM / ZEIT</td>
<td>BMI_S010061_DATUM</td>
</tr>
<tr>
<td>S010061_GRUPPE</td>
<td>Bezeichner, um Kohorten, Sub-Kohorten, Einladungswellen u.a. zu unterscheiden</td>
<td>TEXT</td>
<td>BMI_S010061_GRUPPE</td>
</tr>
<tr>
<td>bmi</td>
<td>Körpermassenindex (Body Mass Index, BMI), kg/m²</td>
<td>ZAHL</td>
<td>BMI_bmi</td>
</tr>
<tr>
<td>bmi_cat</td>
<td>Kategorien des BMI, nach Weltgesundheitsorganisation</td>
<td>ZAHL</td>
<td>BMI_bmi_cat</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Code</th>
<th>Bezeichnung</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Untergewicht</td>
</tr>
<tr>
<td>2</td>
<td>Normalgewicht</td>
</tr>
<tr>
<td>3</td>
<td>Pradipositas</td>
</tr>
<tr>
<td>4</td>
<td>Adipositas</td>
</tr>
</tbody>
</table>
List of Tables

<table>
<thead>
<tr>
<th>Table No.</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>4.1</td>
<td>Format of the raw data for the continuous data visualization in LIFE-SDVS</td>
<td>26</td>
</tr>
<tr>
<td>4.2</td>
<td>Format of the raw data for the categorical data visualization in LIFE-SDVS</td>
<td>27</td>
</tr>
<tr>
<td>4.3</td>
<td>Proportions of population size (in %) in three age groups in three Leipzig Ortsteile in 2013</td>
<td>28</td>
</tr>
<tr>
<td>4.4</td>
<td>Proportions of each age group of Leipzig population in three gender categories</td>
<td>29</td>
</tr>
<tr>
<td>5.1</td>
<td>Frequency (number of Ortsteile) where each candidate of LPA has been selected as best labeling position in Leipzig map</td>
<td>36</td>
</tr>
<tr>
<td>5.2</td>
<td>First five rows of the polygon data of OT75 sorted in ascending order by column y_diff</td>
<td>37</td>
</tr>
<tr>
<td>6.1</td>
<td>Seven main R scripts of LIFE Shiny Application and their major functions</td>
<td>49</td>
</tr>
<tr>
<td>7.1</td>
<td>Partial data output of the function get_group_data_con in the continuous data pre-processing</td>
<td>56</td>
</tr>
<tr>
<td>7.2</td>
<td>Means of hand grip strength in each age group for three gender categories in Leipzig Schleußig and Eutritzsch</td>
<td>60</td>
</tr>
<tr>
<td>7.3</td>
<td>BMI and standard weight status categories classified by World Health Organization</td>
<td>62</td>
</tr>
<tr>
<td>7.4</td>
<td>Percentage of population in each BMI category for two age groups in Germany and in Leipzig</td>
<td>63</td>
</tr>
<tr>
<td>7.5</td>
<td>Example of aggregated data as input for the visualization of categorical data type in LIFE-SDVS</td>
<td>63</td>
</tr>
</tbody>
</table>
List of Figures

1.1 Spot-map by John Snow showing the clusters of cholera cases in London outbreak in 1854 .................................................. 6

2.1 Target sample sizes of the LIFE-Adult-Study ................................................. 10
2.2 An example showing early spatial data visualization in LIFE .......................... 13
2.3 The relative desirability of a set of potential label positions ......................... 15
2.4 Labeling space of an edge ............................................................................. 16
2.5 An example of automatic area-feature labeling result based on general accepted practice in cartography ............................................ 17
2.6 Example of the erosion step of the algorithm by Dörschlag et al. .................... 18
2.7 Examples of leader types in boundary labeling ............................................. 18

3.1 Three-layered architecture of LIFE Spatial Data Visualization System .......... 21
3.2 The relationship of three-layered architecture and the software components of LIFE Spatial Data Visualization System ............................................. 22

4.1 Data preprocessing pipelines in LIFE-SDVS .................................................. 25

5.1 Border of the Ortsteil Zentrum in city Leipzig ............................................. 31
5.2 Map of Leipzig with IDs of Ortsteile positioned at the centroid points .......... 31
5.3 One-step position adjustments of LPA on OT175 .......................................... 33
5.4 Two-step position adjustments of LPA on OT175 ........................................ 34
5.5 The five labeling position candidates generated by LPA of OT10 ................. 35
5.6 Best labeling positions for each Ortsteil of Leipzig found by LPA ................. 36
5.7 Arguments in the plotting function plot_continuous ...................................... 39
5.8 Map of Leipzig labeled with Ortsteil names at the best labeling positions found by LPA .......................................................... 39
5.9 Design of boundary labeling in LIFE-SDVS ................................................ 41
5.10 Map of Leipzig with boundary labeling plotted using function plot_continuous without the Label Alignment Algorithm .................................. 42
5.11 Map of Leipzig with boundary labeling plotted using function plot_continuous with the application of the Label Alignment Algorithm .................................. 42
5.12 Required input data frames and the output of LPA in LIFE-SDVS ............... 44
5.13 Arguments in the plotting function plot_categorical in LIFE-SDVS ................. 45

6.1 Example codes for reactive programming in Shiny .................................................. 47
6.2 Software composition of LIFE Shiny Application and its relationship to architecture components ................................................................. 48
6.3 Screenshot of the tab Data on continuous data user interface in LIFE Shiny Application ................................................................. 50
6.4 Screenshot of the Filter options available in LIFE Shiny Application ......................... 51
6.5 Tab Map Features for continuous data user interface in LIFE Shiny Application ........ 52
6.6 Screenshot of the tab Label Placement for continuous data user interface in LIFE Shiny Application ......................................................... 53
6.7 The tab Map Features on the categorical data user interface of LIFE Shiny Application ................................................................. 54
7.1 Hand grip strength medians of probands of different gender groups in Leipzig Ortsteile 58
7.2 Hand grip strength medians of female probands of different age groups in Leipzig Ortsteile 59
7.3 Maps of Leipzig showing the means of hand grip strength of both genders in each Ortsteile ................................................................. 61
7.4 Proportions of male participants in each BMI category of different age groups in Leipzig Ortsteile ................................................................. 64
7.5 Proportions of female participants in each BMI category of different age groups in Leipzig Ortsteile ................................................................. 65
References


REFERENCES


Erklärung

Ich versichere, dass ich die vorliegende Arbeit selbständig und nur unter Verwendung der angegebenen Quellen und Hilfsmittel angefertigt habe, insbesondere sind wörtliche oder sinn-gemäße Zitate als solche gekennzeichnet.

Mir ist bekannt, dass Zuwiderhandlung auch nachträglich zur Aberkennung des Abschlusses führen kann.

Leipzig, 31.03.2016