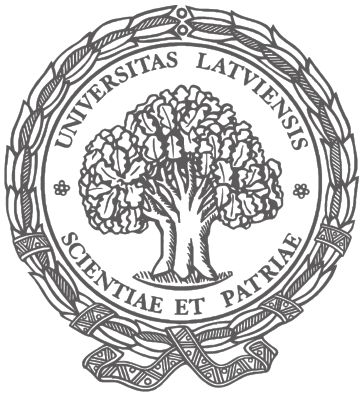
UNIVERSITY OF LATVIA
FACULTY OF PHYSICS AND MATHEMATICS
DEPARTMENT OF MATHEMATICS



THE OPTIMISATION OF SAMPLING DESIGN

DOCTORAL THESIS

Author: **Mārtiņš Liberts**
Student identity card No.: ml07071
Doctoral thesis supervisor: Dr. habil. math., professor Aleksandrs Šostaks
Doctoral thesis adviser: Dr. math. Jānis Lapiņš

RIGA 2013

This work has been supported by the European Social Fund within the project «Support for Doctoral Studies at University of Latvia – 2».

Abstract

The aim of sample surveys is to obtain sufficiently precise estimates of population parameters with low cost. The expected precision of estimates and the expected data collection cost are usually unknown making the choice of sampling design a complicated task. Analytical methods can not be used often because of the complexity of the sampling design or data collection process. The aim of this thesis is to develop a mathematical framework to compare sampling designs of interest with respect to their expected precision of estimates and data collection cost. As a result a framework is developed, which employs artificial population data generation, survey sampling techniques, survey cost modelling, Monte Carlo simulation experiments and other techniques. The framework is applied to analyse the cost efficiency of the Labour Force Survey.

Key words: cost efficiency; simulation study; survey cost estimation; survey methodology; variance estimation.

Mathematics Subject Classification (2010): 62D05.

Anotācija

Izlases apsekojumu mērķis ir iegūt pietiekami augstas precizitātes populācijas parametru novērtējumus ar iespējami mazām izmaksām. Izlases dizaina izvēle parasti ir sarežģīts uzdevums, jo sagaidāmā novērtējumu precizitāte un sagaidāmās datu vākšanas izmaksas nav zināmas. Analītiskas metodes bieži nav iespējams izmantot izlases dizaina sarežģītības vai datu vākšanas procesa sarežģītības dēļ. Promocijas darba mērķis ir izstrādāt matemātisku aparātu, kas ļauj salīdzināt interesējošus izlases dizainus pēc sagaidāmās novērtējumu precizitātes un datu vākšanas izmaksām. Izstrādātais aparāts izmanto mākslīgu populācijas datu ģenerēšanu, izlases apsekojumu metodoloģiju, apsekojuma izmaksu modelēšanu, Monte Karlo simulāciju eksperimentus un citas metodes. Aparāts ir pielietots Latvijas Darbaspēka apsekojuma izmaksu efektivitātes analīzei.

Atslēgvārdi: apsekojumu izmaksu novērtējums; izlases apsekojumu metodoloģija; izmaksu efektivitāte; novērtētāju dispersija; simulāciju eksperiments.

Matemātikas disciplīnu klasifikācija (2010): 62D05.

Contents

Nomenclature **7**

Introduction **8**

List of Publications **13**

List of Presentations **15**

1 Redesign of LFS **17**
1.1 Target Population and Parameters of Interest 17
1.2 Sampling Frame 19
1.3 Definition of the Sampling Design 20
1.3.1 Rotation of Dwellings and Areas 20
1.3.2 Sampling of Areas 21
1.3.3 Rotation Speed 25
1.3.4 Selection of the Design Parameters 25
1.4 Implementation and Properties of the Design 27

2 Artificial Population **29**
2.1 Static Population 29
2.1.1 Statistical Household Register data 29
2.1.2 Labour Force Survey Data 29
2.1.3 Data Merge 31
2.2 Dynamic Population 31
2.2.1 Theoretical Model 34
2.2.2 Estimation of Transition Matrices 34
2.2.3 Generation of Dynamic Population 37

3 Cost Efficiency – Theoretical Model **39**
3.1 Definition of Cost Efficiency 39
3.2 Aim of the Research 40
3.3 Sampling Designs 40

3.3.1 mSRS Design . . . 41
3.3.2 mSSRS Design . . . 42
3.3.3 Choice of Sampling Designs for the Cost Efficiency Study . . . 42
3.4 Estimators of Population Parameters . . . 44
3.5 Variance under mSRS Sampling Design . . . 45
3.5.1 Variance for Estimator of Total . . . 45
3.5.2 Variance for Estimator of Ratio . . . 50
3.5.3 Variance under mSSRS . . . 51
3.6 Estimates by Monte Carlo Simulation . . . 52
3.6.1 Normally Distributed Variable . . . 53
3.6.2 Unspecified Distribution of Variable . . . 53
3.6.3 Nonparametric Bootstrap . . . 54

4 Cost Efficiency – Practical Application 55
4.1 Fieldwork Cost Estimation . . . 55
4.2 Procedures for Monte Carlo Simulations . . . 58
4.3 Cost of Two-Stage Sampling Design . . . 59
4.3.1 Information Available from the Real LFS . . . 59
4.3.2 Estimation of Field Work Budget . . . 59
Simulation of Distance . . . 60
Simulation of Interview Cost . . . 61
Fieldwork Cost . . . 63
4.4 Sample Size of the Alternative Designs . . . 66
4.4.1 Expected Cost and Sample Size . . . 66
4.4.2 Sample Size Estimation . . . 67
4.5 Precision of Population Parameter Estimates . . . 70
4.6 Results of Cost Efficiency Analysis . . . 70

Main Results 80

Acknowledgements 82

References 83

Appendices 86

Appendix 1 R Functions for Monte Carlo Simulations 86
1.1 Sample generation functions . . . 86
1.1.1 SRS . . . 86
1.1.2 SRS by Weeks . . . 87
1.1.3 Cluster Sampling . . . 88

1.1.4 Cluster Sampling by Weeks 90
1.1.5 Stratified Cluster Sampling 91
1.1.6 Two-Stage Sampling 93
1.2 Calculation of Interviewing Expenses 98
1.2.1 TSP Solver for a Single Interviewer 98
1.2.2 TSP Solver for a Multiple Interviewers and Weeks 99
1.2.3 Calculation of Interviewing Expenses 103
1.3 Estimation of Population Parameters 104
1.3.1 Estimation of The Primary Population Parameters 104
1.3.2 Estimation of The Secondary Population Parameters 106
1.4 Other Functions 107
1.4.1 Extraction of Data From The Dynamic Population Data 107
1.4.2 Monte Carlo Simulations 108

Nomenclature

$\doteq$	Approximation
frac (x)	The fractional (non-integer) part of a real number x
$\lfloor x \rfloor$	The integer part of a real number x
$\odot$	Hadamard product
$\boldsymbol{X}$	Matrix
$\boldsymbol{x}$	Vector
M	The total number of units in a set V
N	The total number of elements in a set U
U	The set of all elements over W weeks
u_k	An element k of a set U
U_w	The set of all elements in week w
$u_{i,w}$	An element i of a set U_w
V	The set of all units
v_i	A unit i of a set V
W	The total number of weeks observed

Introduction

The inspiration for this thesis comes from pure practical necessity. National Statistical Institutes (NSIs) are the main providers of official statistics in most countries. A large proportion of official statistics produced by NSIs are done so using data collected via sample surveys, with the main customer of official statistics being the general public (or tax payers, in other words). These days, cost efficiency is an essential consideration in all government spending; the question is, *are NSI sample surveys cost efficient?*

There is not a simple answer to the question posed. A sample survey can possess one of many different sampling designs. The simplest sampling designs do not necessarily provide the lowest data collection cost. More complex sampling designs are considered in theory and applied in practice to obtain statistical information with an acceptable precision at a lower cost. In designing a sample survey, the following considerations should be decided upon: *What is the expected precision of the estimates of population parameters? What is the expected data collection cost? Which sampling design should be chosen in order to minimise sampling errors under a fixed data collection cost?* These are commonly asked questions during the planning stage of a sample survey. In most cases, the answers to the questions posed cannot be gained through analytical means and NSIs are usually reliant on expert's judgement to some extent.

The relation between the precision of estimates and survey cost has been discussed in literature for at least 70 years, though the topic has not been comprehensively addressed by any one author. Different aspects of the relationship have been analysed and different goals of analysis have been set by authors but it is possible to observe the lack of common foundations for the topic.

One of the first sources relating to survey sampling, and discussing the relation between the precision of estimates and survey cost, is a paper by Mahalanobis (1940). The paper is devoted to sample survey methodology, applied to measuring the size of agricultural area used for growing jute in Bengal. The author introduces a cost function to survey a zone (a primary sampling unit here). Different components of the cost function are discussed: enumeration, journey, miscellaneous and indirect. A variance function is introduced and one of the aims of the paper is to minimise the variance function under the fixed cost of the survey.

Another early example is a report by Jessen (1942). It describes a sample survey methodology applied to the case of farm surveys in Iowa. A cost function for the survey is given, whereby the cost function is built on variables such as number of farms in a sampling unit, time spent

on a farm, salary and living expenses of interviewer, average distance between farms within a sampling unit, cost per mile of travel, average speed of travel, number of sampling units in the sample. The problem addressed by the author is to minimise sampling error under a fixed cost.

The topic is discussed extensively in a book by Hansen, Hurwitz, and Madow (1953). Optimal sample size and allocation regarding the fixed cost or the fixed variance in the case of stratified sampling is given in Chapter 5. The construction of an approximate cost function in the case of two-stage cluster sampling is discussed in Section 10 of Chapter 6. The optimisation of design parameters in the case of two-stage cluster sampling is discussed in Section 11 of Chapter 6. The optimal design parameters regarding fixed cost or fixed precision are given. Optimality of design in the case of stratified two-stage cluster sampling and in the case of large primary sampling units is discussed in further chapters. The authors state that survey cost estimation is a complex task:

> The cost function is always difficult to approximate. Often, only a crude approximation can be obtained. A great deal more work and empirical studies and results are needed to improve this phase of the analysis. (Hansen et al., 1953)

All the literature sources mentioned so far in this thesis utilise the same assumption regarding the approximation of travel distance. The travel distance is approximated by $C\sqrt{n}$ where C is some unknown constant and n is the number of sampled primary sampling units. The approximation works well for large n. The result is proven later by Beardwood, Halton, and Hammersley (1959). The problem is that C is unknown and it can vary a great deal, even between different cases within the same survey. The authors provide their own estimates of C but methodology for the estimation of C is not given. It is advised to estimate C from other, previously undertaken, similar surveys or by expert judgement.

Comparison of different sampling designs regarding cost efficiency is given by Kish (1965) in Section 8.3 “Models of Cost Functions”. Practical advice is given by the author regarding the choice of sampling design, with the aim of cost efficiency. A measure of the economy of two sampling designs is introduced. The general cost function for an arbitrary sampling design with four factors is given. However, the author admits to the problem that the factors of the cost function are unknown usually:

> Ordinarily the sampler has no precise data on cost factors, and must base his decisions on estimates or guesses. Often he can make good enough guesses to eliminate designs that would be obviously uneconomical. (Kish, 1965)

A significant book regarding this topic is by Groves (1989). The author starts the discussion with criticism of applied cost-error modelling. Very often, the cost is approximated by linear, continuous and deterministic functions. The author claims that this approach could lead to wrong optimisation results in practice since in reality cost functions tend to be non-linear, discontinuous

and with stochastic features. Non-sampling errors should also be taken into account in cost-error modelling, if possible. The author admits to the fact that closed form analytical solution to optimisation problems may not exist if complexity of cost-error model is increased. The author advocates simulation studies as the best approach for design analysis:

> With complex models, various optimisation problems can be approached with large-scale computer simulation. These simulations can be used to address the common design decisions of the survey statistician – optimal allocation to strata in the sample selection, optimal workloads for interviewers, and optimal number of waves in a panel design. The solutions will be found within the constraints implied by the total budget for a survey. Since it is likely that closed form-solutions to such problems will not exist with complex cost and error models, simulation approaches will be useful to measure the sensitivity of results to changes in various design, cost, or error parameters. (Groves, 1989)

The optimality of cost or precision for the estimation of one population parameter is a classic problem. The selection of optimal sampling designs for multiple-objective surveys is discussed by Malec (1995). Other recent papers regarding cost efficiency of sampling design are Kalsbeek, Botman, Massey, and Liu (1994), Heeringa and Groves (2006) and Mohl and Laflamme (2007).

Several events have been organised recently, in the United States of America, devoted to the topics of survey cost estimation and simulation models for survey fieldwork operations. For example "Survey Cost Workshop" (2006) and "Workshop on Microsimulation Models for Surveys" (2011). Both workshops were organised by the National Institute of Statistical Sciences in Washington, D.C. Several papers devoted to the topic have also been presented in other events such as "Joint Statistical Meeting" (2008) by the American Statistical Association, "Research Conference" (2012) by the Federal Committee on Statistical Methodology and the "Statistics and Public Policy Workshop" (2012) by the American Statistical Association. The research of survey field operations is a brand new topic in the scope of statistical research. Several research activities have been devoted to the topic only recently. Two quotes follow:

> So far, similar work is rarely found in the literature describing the analytical or simulation modeling of the operations. The field operation is a unique system in the operations research field. (Chen, 2008)

> The field operations of surveys can be classified as stochastic dynamic systems. Usually the field operations cannot be modelled analytically because of the complexity of the system. (Cox, 2012)

This literature review concludes with a very recent paper by Calinescu, Bhulai, and Schouten (2013). The paper aims to solve the resource allocation problem in survey designs using adaptive sampling design and operational research techniques:

> Resource allocation is a relatively new research area in survey designs and has not been fully addressed in the literature. Recently, the declining participation rates and increasing survey costs have steered research interests towards resource planning. Survey organizations across the world are considering the development of new mathematical models in order to improve the quality of survey results while taking into account optimal resource planning. (Calinescu et al., 2013)

Some general observations can be drawn from this literature review. In general, the total price for a survey, where data are collected directly from respondents, is increasing. There are several reasons for the increase of the price, but one significant reason is decreasing response. In today's world, either much more effort is needed to increase the cost efficiency of surveys, or a higher price must be paid, in order to produce the same quality of statistics as in times when non-response was not such a big problem. However, given the current economic climate, in most cases it is simply not possible to spend more since most government budgets for surveys are reducing, or at best being kept the same as the previous year's. It is clear, therefore, that increased cost efficiency is crucial to maintaining the production of high quality statistics under a decreasing or fixed budget. Since survey sampling emerged as a methodology, problem with non-response and budget restrains has not been met so often. This is one of the main reasons why survey cost efficiency has not been a very important research topic until recently.

Another observation is that, simulation experiments are getting more and more attention as a tool used in the designing of production systems for official statistics. The expansion of the method is possible because of a cheap computer power available currently, even a desktop or a laptop computer nowadays can be set up to solve large scale simulation experiments.

The Latvian Labour Force Survey (LFS) is the main object of the study in the thesis. It was organised for the first time in November 1995 (Lapiņš, 1997) and ran biannually. The first redesign of the LFS sampling design was done after the 2000 Latvian Population Census with the new sampling design launched in 2002 (Lapiņš, Vaskis, Priede, & Bāliņa, 2002). It become a continuous survey after the redesign. The second redesign of the survey occurred in 2006. The re-launch of the LFS with the new sampling design and a much larger sample size took place in 2007. Finally, the latest redesign of the LFS sampling design was done in the scope of this thesis by the author in 2009 (Liberts, 2010a). The main reason for redesigning the LFS sampling design was the necessity to update the population frame used for the first-stage sampling units. The redesign resulted with a new sample drawn which was used to run the LFS since 2010. More information regarding the history of the LFS is given by Central Statistical Bureau of Latvia (2012a) and European Commission (2012a, 2012b).

The goal of this thesis is to develop a framework which can be used to compare arbitrary sampling designs by their cost efficiency. The framework should be used to analyse selected sampling designs and determine the sampling design that leads to the highest overall precision of estimates under a fixed survey budget. The following tasks are set to achieve the goal:

1. to update the frame of primary sampling units and implement the redesign of the LFS,

2. to create artificial population data representing the statistical characteristics of the target population of the LFS,
3. to compare the sampling design of the LFS with alternative sampling design with respect to the cost efficiency using the developed framework,
4. to provide recommendations for the choice of the LFS sampling design with respect to the cost efficiency.

The first chapter of the thesis is devoted to the redesign of the LFS in 2009. The LFS sampling design is defined and the process of the redesign is described. The resulting sampling design is the basis for the analysis of the thesis. Methodology to develop artificial population data is presented in the second chapter. Artificial population data with characteristics similar to the target population of the LFS have been produced with this methodology. The third chapter of this thesis is devoted to the theoretical development of the framework for the cost efficiency analysis. The application of the framework is presented in the fourth chapter of this thesis. The appendix of this thesis contains the description and code of the developed R procedures used to achieve the results of this thesis.

List of Publications

Publications

Liberts, M. (2010a). The redesign of Latvian Labour Force Survey. In M. Carlson, H. Nyquist, & M. Villani (Eds.), *Official statistics – methodology and applications in honour of Daniel Thorburn* (pp. 193–203). Stockholm, Sweden: Stockholm University. Retrieved from `http://officialstatistics.wordpress.com/`

Liberts, M. (2010b). The weighting in household sample surveys. In O. Krastiņš & I. Vanags (Eds.), *The results of statistical scientific research 2010* (pp. 168–174). Riga: Central Statistical Bureau of Latvia. Retrieved from `http://home.lu.lv/~pm90015/work/PhD/pub/10Papers/Liberts_2010_Weighting.pdf`

Liberts, M. (2013). *The cost efficiency of sampling designs.* Manuscript submitted for publication in the journal *Statistics in Transition – new series*.

Report

Liberts, M. (2010). Country report – Latvia. *The Survey Statistician*, *62*, 19–20. Retrieved from `http://isi.cbs.nl/iass/survstatUK.htm`

Conference Theses

Liberts, M. (2010a, April). Self-rotating sampling design. In *Abstracts of 8th Latvian mathematical conference* (p. 46). Valmiera, Latvia. Retrieved from `http://home.lu.lv/~pm90015/work/PhD/pub/30ConfTh/LMB8_MLiberts.pdf`

Liberts, M. (2010b, June). Self-rotating sampling design. In *Abstracts of 10th international Vilnius conference on probability theory and mathematical statistics* (p. 212). Vilnius, Lithuania: TEV Publishers. Retrieved from `http://home.lu.lv/~pm90015/work/PhD/pub/30ConfTh/IVC2010_MLiberts.pdf`

Liberts, M. (2011, June). Simulation study of sampling design in Labour Force Survey. In *Proceedings of third Baltic-Nordic conference in survey statistics* (p. 52).

Norrfällsviken, Sweden. Retrieved from http://www.mathstat.helsinki.fi/msm/banocoss/2011/Presentations.html

Liberts, M. (2012, June). Survey design analysis regarding cost efficiency. In *Programme of nordstat 2012, 24th Nordic conference in mathematical statistics* (pp. 47–48). Umeå, Sweden. Retrieved from http://www.trippus.se/eventus/userfiles/33956.pdf

Workshop Theses

Liberts, M. (2010, August). Weighting and estimation in household surveys with rotating panel. In *Proceedings of Baltic-Nordic-Ukrainian workshop on survey sampling theory and methodology* (pp. 9–10). Vilnius, Lithuania: Statistics Lithuania. Retrieved from http://vilniusworkshop2010.stat.gov.lt/Scientific%20programme.html

Liberts, M. (2012, August). The simulation study of survey cost and precision. In *Lecture materials and contributed papers of workshop of Baltic-Nordic-Ukrainian network on survey statistics* (pp. 124–128). Valmiera, Latvia: University of Latvia, Central Statistical Bureau of Latvia. Retrieved from http://home.lu.lv/~pm90015/workshop2012/speakers.shtml

Seminar Thesis

Liberts, M. (2011, April). *Some aspects on simulation study of a sampling design.* Joint statistical seminar at Umeå University. Umeå, Sweden. Retrieved from http://www.usbe.umu.se/enheter/stat/forskning/seminarier/varen-2011

List of Presentations

Lecture

Liberts, M. (2010, August). *Weighting and estimation in household surveys with rotating panel.* Baltic-Nordic-Ukrainian Workshop on Survey Sampling Theory and Methodology. Vilnius, Lithuania. Retrieved from http://home.lu.lv/~pm90015/work/PhD/pub/50Lect/BNU2010_MLiberts.pdf

Presentations at Conferences

Liberts, M. (2008, February). *Divpakāpju izlases dizaina plānošana.* The 66th Scientific Conference of the University of Latvia. Riga, Latvia. Retrieved from http://home.lu.lv/~pm90015/work/PhD/pub/60PresConf/LU_2008_MLiberts.ppt (in Latvian)

Liberts, M. (2010a, March). *Izlases dizaina optimizācija.* The 68th Scientific Conference of the University of Latvia. Riga, Latvia. Retrieved from http://home.lu.lv/~pm90015/work/PhD/pub/60PresConf/LU_2010_MLiberts.pdf (in Latvian)

Liberts, M. (2010b, April). *Pašrotējošs izlases dizains.* 8th Latvian Mathematical Conference. Valmiera, Latvia. Retrieved from http://home.lu.lv/~pm90015/work/PhD/pub/60PresConf/LMK8_MLiberts.pdf (in Latvian)

Liberts, M. (2010c, June). *Self-rotating sampling design.* 10th International Vilnius Conference on Probability Theory and Mathematical Statistics. Vilnius, Lithuania. Retrieved from http://home.lu.lv/~pm90015/work/PhD/pub/60PresConf/IVC2010_MLiberts.pdf

Liberts, M. (2011, June). *Simulation study of sampling design in Labour Force Survey.* Third Baltic-Nordic Conference in Survey Statistics. Norrfällsviken, Sweden. Retrieved from http://home.lu.lv/~pm90015/work/PhD/pub/60PresConf/BaNoCoSS_MLiberts.pdf

Liberts, M. (2012, June). *Survey design analysis regarding cost efficiency.* Nordstat 2012, 24th Nordic Conference in Mathematical Statistics. Umeå, Sweden. Retrieved from http://arthur.math.umu.se/Nordstat2012/Presentations/C12/MLiberts.pdf

Liberts, M. (2013, February). *Izmaksu efektivitātes novērtēšana izlases apsekojumiem.* The 71st Scientific Conference of the University of Latvia. Riga, Latvia. Retrieved from `http://home.lu.lv/~pm90015/work/PhD/pub/60PresConf/LU_2013_MLiberts.pdf` (in Latvian)

Presentations at Workshops

Liberts, M. (2010, April). *Self-rotating sampling design in Latvian LFS.* Workshop on labour force survey methodology. Paris, France. Retrieved from `http://ej.uz/wlfsm` (organised by Eurostat and French National Institute for Statistics and Economic Studies)

Liberts, M. (2012, August). *The simulation study of survey cost and precision.* Workshop of Baltic-Nordic-Ukrainian network on survey statistics. Valmiera, Latvia. Retrieved from `http://home.lu.lv/~pm90015/workshop2012/speakers`

Presentations at Seminars and Other Presentations

Liberts, M. (2010, September). *Survey sampling methodology in the official statistics of Latvia.* Weekly seminar of Institute of Mathematical Statistics. Tartu, Estonia. Retrieved from `http://home.lu.lv/~pm90015/work/PhD/pub/80PresSem/UT2010_MLiberts.pdf` (during a study visit at Institute of Mathematical Statistics, Faculty of Mathematics and Computer Science, Tartu University)

Liberts, M. (2011, April). *Some aspects on simulation study of a sampling design.* Joint statistical seminar at Umeå University. Umeå, Sweden. Retrieved from `http://home.lu.lv/~pm90015/work/PhD/pub/80PresSem/UU2011_MLiberts.pdf` (during a study visit at Department of Mathematics and Mathematical Statistics, Umeå University)

Liberts, M. (2012a, February). *Izlases dizaina optimizācija: problemātika un daži rezultāti.* Scientific Workshop in Mathematical Statistics at University of Latvia. Riga, Latvia. Retrieved from `http://home.lu.lv/~valeinis/lv/seminars` (in Latvian)

Liberts, M. (2012b, November). *The optimisation of sampling design.* Presentation of the doctoral thesis to Dr.habil.math. Aivars Lorencs (a senior researcher at the Institute of Electronics and Computer Science). Riga, Latvia. (in Latvian)

Liberts, M. (2012c, December). *The optimisation of sampling design.* Presentation of the doctoral thesis at the meeting of Mathematical Analysis Chair, University of Latvia. Riga, Latvia. Retrieved from `http://home.lu.lv/~pm90015/work/PhD/pub/90OthPres/LU_Katedras_sede_28Dec.pdf` (in Latvian)

Chapter 1

Redesign of LFS

The chapter presents the two-stage sampling design used for the Latvian Labour Force Survey (LFS). It is based on the publication by Liberts (2010a) unless otherwise stated. The publication presents methodology for the third redesign of the LFS. The redesign was initialized in 2009 and methodology for the new sampling design was developed by the author.

The main reason for redesigning the LFS sample was the necessity to update the population frame used for the first-stage sampling units. The population frame used for the first-stage sampling was the the list of census-counting areas – a list which had not been updated since the population census in 2000. A study analysing the coverage of the first-stage population frame revealed that over-coverage was around 1.4% and under-coverage 2.2%. The study also showed that the size of the primary sampling units (PSUs) was outdated. The largest discrepancies were found in rural areas and in the capital city – Riga. For these reasons, the decision was made to update the first-stage population frame using the latest available information.

The new sampling design is already implemented for the LFS and other surveys and is used since 2010. It is the basis for the further studies in this thesis. Survey sampling theory is used to provide the results of this chapter.

1.1 Target Population and Parameters of Interest

The target population of LFS is defined to be the residents of Latvia permanently living in private households. Residents of working age (15–74 year old) compose the main domain of interest. It is important to note that the characteristics of the target population is continuously changing over time, for example, there are individuals who are gaining jobs and loosing jobs every day. The target population is observed on a weekly bases using the LFS methodology (European Commission, 2012b, p. 5). All residents of Latvia would therefore have to be questioned every week if the LFS would be done as a census[1] to measure the population parameters necessary to produce.

[1] a full survey of the whole target population

An individual is called a **unit** and denoted by v_i (there are cases when households are used as units). The set of all units is denoted by V. The size of V is M, $|V| = M$. The units are labelled with an index i where $i \in (1, M)$, $V = \{v_1, v_2, \ldots, v_M\}$.

The observation of a unit v_i in week w is called an **element** and is denoted by $u_{i,w}$. The set of all elements in week w is denoted by U_w. There are M elements in a set U_w, $|U_w| = M$. The elements of U_w are labelled with a double index (i, w), where i refers to the unit observed and w refers to the week observed, $U_w = \{u_{1,w}, u_{2,w}, \ldots, u_{M,w}\}$. A value $y_{i,w}$ is associated with each element $u_{i,w}$ from the set U_w. The total of a variable y in week w is defined as

$$Y_w = \sum_{i=1}^{M} y_{i,w},$$

and the variance of a variable y in week w is defined as

$$S_w^2 = \frac{1}{M-1}\left(\sum_{i=1}^{M} y_{i,w}^2 - \frac{1}{M}Y_w^2\right).$$

The number of weeks observed is denoted by W and w is the week index, $w \in \overline{1, W}$. The set of elements over W weeks is denoted by U, $U = \cup_{w=1}^{W} U_w$. Each U_w consists of the same units from V but observed in different weeks. The size of U_w is constant over time, $|U_w| = M$ for all w. The size of U is denoted by N, $|U| = \sum_{w=1}^{W} M = WM = N$. An index k is used to label elements over W weeks, $k \in \overline{1, N}$. The elements of each U_w are ordered according to the order of units in V. Indices $\{k : ((k-1) \bmod M) + 1 = i\}$ correspond to a unit v_i. The total of a variable y over W weeks is defined as

$$Y = \sum_{w=1}^{W} Y_w = \sum_{w=1}^{W}\sum_{i=1}^{M} y_{i,w} = \sum_{k=1}^{N} y_k,$$

and the covariance of variable y in weeks w and v is defined as

$$S_{w,v} = \frac{1}{M-1}\left(\sum_{i=1}^{M} y_{i,w}y_{v,i} - \frac{1}{M}Y_wY_v\right).$$

An illustration of units and elements is given in Table 1.1. The rows of the table represent the units. There are M rows. The columns of the table represent the weeks observed. There are W columns. The cells of the table represent elements. The dimension of the table is $M \times W$.

Two types of population parameter are considered for estimation – quarterly average of weekly totals and quarterly ratio of two totals. It is assumed there are 13 weeks each quarter.

Table 1.1

The illustration of units and elements

i	$w=1$	$w=2$	$w=3$	$w=4$	$w=5$	$\cdots$	$w=W$
1	$u_{1,1}$	$u_{1,2}$	$u_{1,3}$	$u_{1,4}$	$u_{1,5}$	$\cdots$	$u_{1,W}$
2	$u_{2,1}$	$u_{2,2}$	$u_{2,3}$	$u_{2,4}$	$u_{2,5}$	$\cdots$	$u_{2,W}$
3	$u_{3,1}$	$u_{3,2}$	$u_{3,3}$	$u_{3,4}$	$u_{3,5}$	$\cdots$	$u_{3,W}$
4	$u_{4,1}$	$u_{4,2}$	$u_{4,3}$	$u_{4,4}$	$u_{4,5}$	$\cdots$	$u_{4,W}$
5	$u_{5,1}$	$u_{5,2}$	$u_{5,3}$	$u_{5,4}$	$u_{5,5}$	$\cdots$	$u_{5,W}$
6	$u_{6,1}$	$u_{6,2}$	$u_{6,3}$	$u_{6,4}$	$u_{6,5}$	$\cdots$	$u_{6,W}$
$\cdots$	$\cdots$	$\cdots$	$\cdots$	$\cdots$	$\cdots$	$\cdots$	$\cdots$
M	$u_{M,1}$	$u_{M,2}$	$u_{M,3}$	$u_{M,4}$	$u_{M,5}$	$\cdots$	$u_{M,W}$

The weekly total of a variable y for week w is defined by

$$Y_w = \sum_{i=1}^{M} y_{i,w},$$

and the quarterly average of the weekly totals of a variable y is defined by

$$Y_q = \frac{1}{13}\sum_{w=1}^{13} Y_w = \frac{1}{13}\sum_{w=1}^{13}\sum_{i=1}^{M} y_{i,w} = \frac{1}{13}\sum_{k=1}^{N} y_k. \tag{1.1}$$

The quarterly ratio of two totals is defined by

$$R_q = \frac{Y_q}{Z_q} = \frac{\sum_{w=1}^{13} Y_w}{\sum_{w=1}^{13} Z_w} = \frac{\sum_{w=1}^{13}\sum_{i=1}^{M} y_{i,w}}{\sum_{w=1}^{13}\sum_{i=1}^{M} z_{i,w}} = \frac{\sum_{k=1}^{N} y_k}{\sum_{k=1}^{N} z_k}. \tag{1.2}$$

1.2 Sampling Frame

There is the Population Register in Latvia (*Population Register Law*, 1998). The Central Statistical Bureau (CSB) receives the data from the Population Register continuously. The data are received in different formats:

- There is online access to the register – mainly used for data checking;
- There are data received monthly – mainly used for demographic statistics and the updating of the Statistical Household Register.

The data from the register are kept, edited and matched with other information by the CSB in the Statistical Household Register. The basic unit in the data is the individual. Individuals are merged to construct dwellings. Individuals are merged by declared address of living or other information. Dwellings are used as secondary sampling units for most of the household surveys organised by the CSB. Dwellings are merged to construct census-counting areas – PSUs for the household surveys.

The first task was to update the population frame of PSUs which in our case are census counting areas. The following changes were made. There were dwellings in the Statistical Household Register not matched to any census counting area. There were 2.1% such dwellings in total (4.1% in rural areas). Previously those dwellings were not included in sampling frame and they amounted to under-coverage.

The State Land Service of the Republic of Latvia is an institution that holds information about buildings in Latvia. The information about the buildings includes the geographical coordinates of the building. This information is available to the CSB. Geographical coordinates were used to match dwellings to PSUs. The coordinates for some buildings were not known in some cases. Coordinates of the street, parish or territorial unit were used in those cases. All dwellings were matched to PSUs in the end.

The size of the PSUs was recalculated. The size was defined as number of private dwellings in the area. There were some rural areas where the addresses of dwellings were not defined. The registered address of such dwellings was the name of the parish or other area. Such imprecisely defined addresses can contain many dwellings and individuals. A new algorithm was created to approximately define dwellings based on the surname and farm identification.

There was a reform of administrative territories in Latvia. PSUs were reordered to form a serpentine shape by the CSB. The ordering of the areas was done in each stratum separately. The areas were ordered so that successive areas in the ordered list were geographically close to each other. The areas formed a closed shape. The census counting areas, ordered in serpentine shape, are shown in Figure 1.1. Geographical coordinates, using the coordinate system LKS92 (*Ģeodēziskās atskaites sistēmas un topogrāfisko karšu sistēmas noteikumi*, 2011) are used in the figure.

The size of some PSUs was too small – there were not enough addresses to be selected for the sample. For each stratum the minimum PSU size was defined. PSUs below minimum size were merged with other PSUs from the same territorial area.

Geographical coordinates were computed for the centre of each PSU. The centre was defined by the location of dwellings in the area. If the area was too small, it was merged with the closest area of the same territorial unit. The closest area was defined by the shortest distance between the centres of areas.

1.3 Definition of the Sampling Design

1.3.1 Rotation of Dwellings and Areas

The rotation scheme `2-(2)-2` is implemented for dwellings in the LFS sampling design. According to European Commission (2012b), it is a scheme, "where sampled units are interviewed for two consecutive quarters, than stay out of the sample for the next two quarters and are included again two more times afterwards" (p. 7). Areas are rotated by rotation scheme `8-()`,

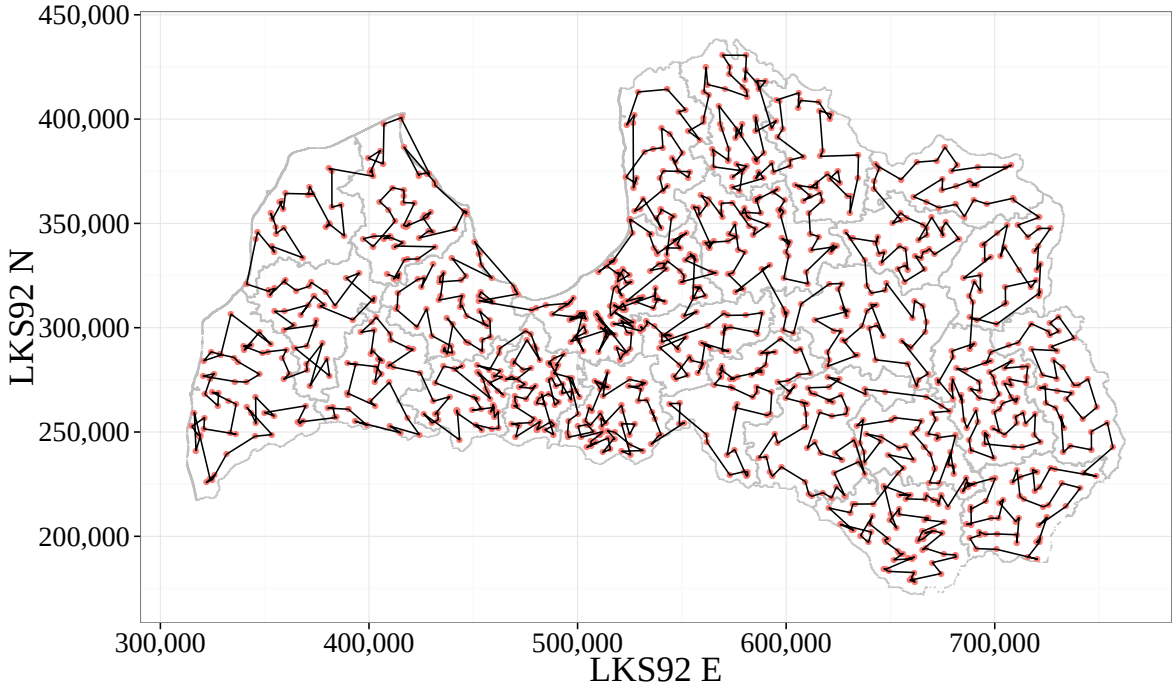


Figure 1.1 **PSUs of rural areas ordered in serpentine shape**

Table 1.2

Scheme of Rotation of Dwellings in a Sampled Area

Sample	q1	q2	q3	q4	q5	q6	q7	q8
Sample 1	1	2	·	·	3	4	·	·
Sample 2	·	·	1	2	·	·	3	4

where areas are sampled for eight consecutive quarters and then rotated out of the sample. Two different non-overlapping samples of dwellings are selected in each sampled area. The first sample of dwellings is used in the 1st, 2nd, 5th and 6th quarter (the first row). The second sample of dwellings is used in 3rd, 4th, 7th and 8th quarter (the second row). The desired LFS rotation scheme `2-(2)-2` is realised for both samples of dwellings. The rotation scheme for dwellings and areas is illustrated in Table 1.2.

1.3.2 Sampling of Areas

A stratified systematic πps sampling design (Särndal, Swensson, & Wretman, 1992) is used to select areas. There are four strata. The strata are defined by the rural-urban classification:

- Riga – the capital city of Latvia,
- Cities under state jurisdiction excluding Riga (8 cities),
- Towns (68 towns),
- Rural areas (512 areas).

A separate sample of areas is selected each week because the LFS is a continuous survey. The weekly sample size of areas is 8 in the first stratum (Riga) and 16 in all other strata.

Assume we have ordered all PSUs from a stratum with their dwellings on a circle. A dwelling is used as a unit v_i here. PSUs are ordered in a serpentine shape, as described in Section 1.2. Assume the length of the circle is equal to 1. The circle represents the set V of one stratum here. Assume all units v_i of one stratum are placed evenly on the circle – the equal sized arcs of the circle represent each dwelling. If we put a point on the circle, we have selected a unit v_i – by selecting a unit v_i we have selected a PSU.

There is an octagon in the circle as shown in Figure 1.2. Evidently the octagon divides the circle in eight arcs. There are seven arcs with a length $\frac{1+\delta_h}{8}$ and an arc – with a length $\frac{1-7\delta_h}{8}$, where δ_h is a chosen constant for each stratum, h is the stratum index, $h = \{1, 2, 3, 4\}$. The number of dwellings in the PSU defines the size of PSU. A statistician can choose the value of δ_h freely. But it is reasonable to choose

$$\frac{\max_i(M_{hi})}{\sum_i(M_{hi})} < \delta_h < \frac{1}{8},$$

where M_{hi} is the size of PSU i in stratum h. We will call δ_h sampling displacement. Evidently

$$7\frac{1+\delta_h}{8} + \frac{1-7\delta_h}{8} = 1.$$

To select the necessary number of areas in a weekly sample a single octagon is needed in stratum "Riga" (sample size 8) and two octagons are needed in each of the other strata (sample size 16). Let $a\,(w, h, o, v)$ denote a point on the circle, where w is the week index, $h \in \{1, 2, 3, 4\}$ is the stratum index, $o \in \{1, 2\}$ is the octagon index, and $v \in \overline{1, 8}$ is the vertex index.

The first point on the circle is selected randomly $a\,(w = 1, h, o = 1, v = 1) = \xi_h$, where ξ_h is distributed uniformly in the interval $(0, 1)$. The point $a\,(w = 1, h, o = 1, v = 1)$ defines a PSU sampled for the eighth time (it will be rotated out of the sample in the next quarter). The second point is computed as $a\,(w = 1, h, o = 1, v = 2) = \text{frac}\left(\xi_h + \frac{1+\delta_h}{8}\right)$. The second point selects a PSU sampled for the seventh time. All vertices of the octagon can be computed by

$$a\,(w = 1, h, o = 1, v) = \text{frac}\left(\xi_h + \frac{v-1}{8}\,(1+\delta_h)\right).$$

The selection time of an area can be computed as $R\,(v) = 9 - v$. The numbers attached to the objects (1-8) represent the selection time of a PSU in the sample (Figure 1.2). Two non-overlapping samples of dwellings are selected in each PSU. The circles represent the first sample of dwellings; the squares represent the second sample of dwellings.

The number of areas already selected is enough for the weekly sample of stratum "Riga". Sampling eight more areas is necessary for all of the other strata. To get more areas we will rotate the current octagon by the length of arc equal to $\frac{9}{16}\,(1+\delta_h)$. The circles and the squares

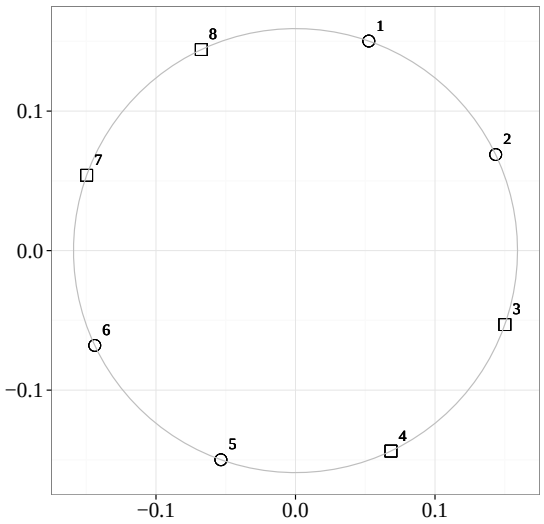


Figure 1.2 **The first octagon**

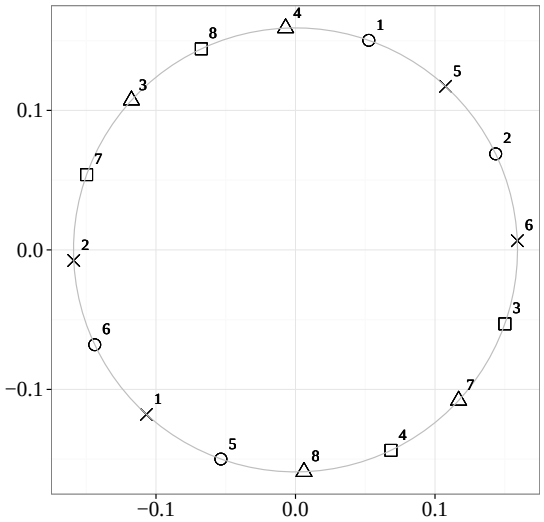


Figure 1.3 **The first and the second octagon**

represent the first octagon; the crosses and the triangles represent the second octagon in Figure 1.3.

All 16 PSUs selected in the sample for the 1st week can be computed by the following numbers (NB only the first eight points are used for the stratum "Riga"):

$$\begin{array}{cc}
\xi_h & \text{frac}\left(\xi_h+\frac{1}{8}\left(1+\delta_h\right)\right) \\
\text{frac}\left(\xi_h+\frac{2}{8}\left(1+\delta_h\right)\right) & \text{frac}\left(\xi_h+\frac{3}{8}\left(1+\delta_h\right)\right) \\
\text{frac}\left(\xi_h+\frac{4}{8}\left(1+\delta_h\right)\right) & \text{frac}\left(\xi_h+\frac{5}{8}\left(1+\delta_h\right)\right) \\
\text{frac}\left(\xi_h+\frac{6}{8}\left(1+\delta_h\right)\right) & \text{frac}\left(\xi_h+\frac{7}{8}\left(1+\delta_h\right)\right) \\
\text{frac}\left(\xi_h+\frac{9}{16}\left(1+\delta_h\right)\right) & \text{frac}\left(\xi_h+\left(\frac{9}{16}+\frac{1}{8}\right)\left(1+\delta_h\right)\right) \\
\text{frac}\left(\xi_h+\left(\frac{9}{16}+\frac{2}{8}\right)\left(1+\delta_h\right)\right) & \text{frac}\left(\xi_h+\left(\frac{9}{16}+\frac{3}{8}\right)\left(1+\delta_h\right)\right) \\
\text{frac}\left(\xi_h+\left(\frac{9}{16}+\frac{4}{8}\right)\left(1+\delta_h\right)\right) & \text{frac}\left(\xi_h+\left(\frac{9}{16}+\frac{5}{8}\right)\left(1+\delta_h\right)\right) \\
\text{frac}\left(\xi_h+\left(\frac{9}{16}+\frac{6}{8}\right)\left(1+\delta_h\right)\right) & \text{frac}\left(\xi_h+\left(\frac{9}{16}+\frac{7}{8}\right)\left(1+\delta_h\right)\right)
\end{array}$$

The 16 points for the first week can be expressed alternatively by

$$a\left(w=1,h,o,v\right)=\text{frac}\left(\xi_h+\left(\frac{B_o}{16}+\frac{v-1}{8}\right)\left(1+\delta_h\right)\right),$$

where $h\in\{1,2,3,4\}$ is the stratum index, $o\in\{1,2\}$ is the octagon index,

$$B_o\in\begin{cases}\{0\}, & \text{if}\quad h\in\{1\},\\ \{0,9\}, & \text{if}\quad h\in\{2,3,4\},\end{cases}$$

and $v\in\overline{1,8}$ is the vertex index.

The sampling step

$$\Delta_h=\frac{q_h}{13}+\frac{1+\delta_h}{8\cdot 13}$$

is computed to select PSUs for the next weeks, where q_h is an integer representing the "speed" of rotation. The position for each point on the circle is computed by

$$a\left(w,h,o,v\right)=\text{frac}\left(\xi_h+\left(\frac{B_o}{16}+\frac{v-1}{8}\right)\left(1+\delta_h\right)+\left(w-1\right)\Delta_h\right). \tag{1.3}$$

It is possible to verify that $a\left(w,h,o,v=8\right)\equiv a\left(w+13,h,o,v=7\right)$. The equality means that the 7th vertex of the octagon is placed in the same position as where the 8th vertex was placed 13 weeks ago. Moreover, the expression $a\left(w+13\left(l-1\right),h,o,v=9-l\right)$ has the same value for all $l=\overline{1,8}$. This means that if we compare the placement of the octagon in week w and week $w+13$ the positions of seven vertices will coincide (the eighth vertex will be shifted by δ_h). In other words, if a PSU is sampled for the first time in week w, then it will be also sampled in weeks $w+13$, $w+26$, $w+39$, $w+52$, $w+65$, $w+78$, $w+91$.

Direct calculations show that $a\left(w+104,h,o,v\right)\equiv\text{frac}\left(a\left(w,h,o,v\right)+\delta_h\right)$. The equality means that if a PSU has already been sampled eighth times (at 13 week intervals) then it will not

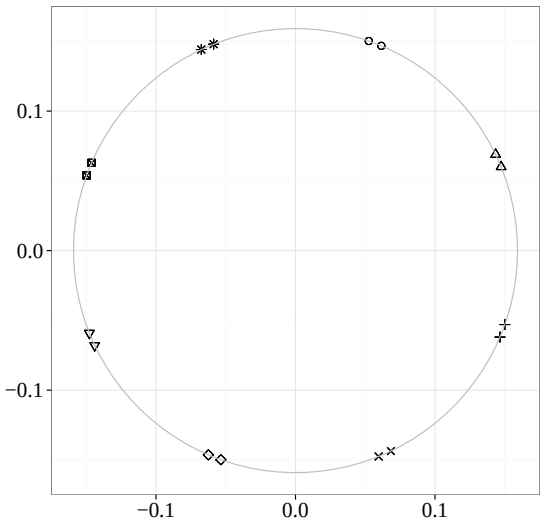


Figure 1.4 **The sample of two weeks**

be sampled the ninth time (104 weeks after the first sampling of the PSU) since the corresponding vertex of the octagon will be shifted by δ_h from its initial position.

1.3.3 Rotation Speed

If $q_h = 0$, the sampled areas of successive weeks are located geographically close to each other. The example of areas sampled during a 13 week period where $q_h = 0$ is shown in Figures 1.4, 1.5 and 1.6. The different shapes of the objects in the figures represent each vertex of the octagon in the figures. The geographical closeness of the sampled areas of successive weeks is not reasonable for several reasons:

- the monthly sample (the sample of four or five succeeding weeks) is not geographically evenly distributed, and it could lead to non-precise estimates of the monthly parameters (NB monthly parameters are not estimated by the Latvian LFS yet),
- the sample could result in an uneven workload for interviewers over time (there is a high probability that some interviewers will have to work for two succeeding weeks if sampled units for both weeks are geographically close to each other).

It is advisable to choose $q_h > 0$. There is no merit in considering $q_h \geq 13$, because only the fractional part of $\frac{q_h}{13}$ influences the positions of the points on the circle. Therefore it is advised to choose q_h where $q_h \in \{1, 2, \ldots, 12\}$. There are 12 possible designs to choose from.

1.3.4 Selection of the Design Parameters

Assume a sample of PSUs is drawn by the new sampling design and is denoted as the new sample. The PSU sample drawn by the previous sampling design is denoted as the old sample. There will be an overlap of the new and the old sample for five quarters because of the longitudinal feature of the LFS. The overlap of the samples is described by Liberts (2010b, p. 170).

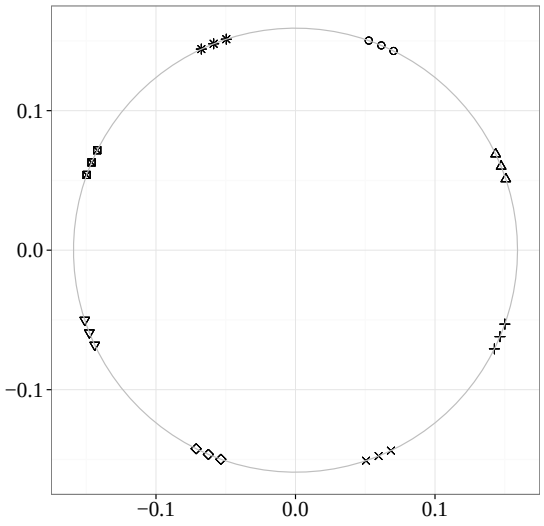


Figure 1.5 **The sample of three weeks**

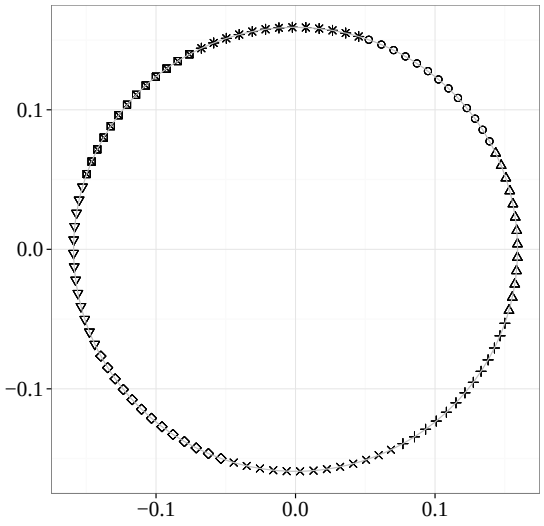


Figure 1.6 **The sample of thirteen weeks**

Table 1.3

Selected design parameters

h	q_h	δ_h
1	6	0.00169
2	2	0.00283
3	7	0.00210
4	11	0.00288

There are two design parameters for the optimisation of the sampling design – q_h and δ_h. The optimisation criteria was – minimal total overlap of the new and the old sample during the five quarter transition period. The size of the overlap for one quarter is defined as the total number of PSUs sampled simultaneously by both samples in a quarter. The sizes of the overlap for each quarter are summed, achieving the size of the total overlap.

A sub-optimal solution for each stratum was found by grid searching through a subset of the parameter space. All 13 valid values for $q_h \in \overline{0, 12}$ and 20 possible values for

$$\delta_h \in \left\{ \frac{10}{10 \max_i(M_{hi})}, \frac{11}{10 \max_i(M_{hi})}, \ldots, \frac{29}{10 \max_i(M_{hi})} \right\}$$

were selected. The subset of the parameter space was created as a Cartesian product from the sets of selected values of both parameters. The size of the resulting subset was equal to 260. The expected size of the overlap was estimated for each pair of the parameter values by simulations. The pair of the parameter values giving the smallest size of overlap was selected as a sub-optimal solution for the problem. The resulting values of parameters are shown in Table 1.3. We believe the sub-optimal solution is close to the optimal solution because all valid values of the parameter q_h were tested and the expected size of overlap is not changing much regarding the changes in δ_h.

1.4 Implementation and Properties of the Design

The following steps are done to draw a sample using the two-stage sampling design described:

- the values of ξ_h are drawn in each stratum h using a uniformly distributed random variables in interval $(1, 0)$,
- the values of q_h and δ_h are selected for each stratum,
- the values of $a\,(w, h, o, v)$ are computed for time period necessary (for example 5 years),
- the values of $b\,(w, h, o, v) = a\,(w, h, o, v)\,M_h$ are computed to select PSU sample, where M_h is a number of dwellings in each stratum (the PSU sample is selected according to the numbers $b\,(w, h, o, v)$ using cumulative summation),

- a simple random sample of dwellings with fixed sample size is drawn from each sampled PSU for the first or the third selection time (a modified simple random sampling method is used usually – the households previously sampled for LFS or other surveys are excluded from the sampling frame).

The sampling design created possesses several statistical and practical properties.

1. The sample drawn by the design is a probability sample according to Särndal et al. (1992, p. 8).
2. It is a self-weighting design in each stratum.
3. The design provides the rotation of the sampled units according to the rotation pattern `2-(2)-2`.
4. There is simple management of PSUs in a sample. The sample of PSUs can be selected for several years in advance. It allows plenty of time for planning the interviewers' work schedules.
5. It is possible to coordinate different continuous household samples. Currently the samples for LFS, Household Budget Survey (HBS) and Survey of Domestic Travellers (SDT) are coordinated with each other. The PSU samples of HBS and SDT are sub-samples of the LFS PSU sample. An interviewer can undertake all three surveys in a PSU with shorter travelling distances compared to uncoordinated sampling design. The coordination allows the total cost of the three surveys to be kept low.
6. It is possible to use the PSU sample selected by the design for other ad-hoc sample surveys.
7. The design is suitable for application of different re-sampling variance estimation methods (for example the method of non-independent random groups or Jackknife (Wolter, 2007)).

Chapter 2

Artificial Population

The chapter presents the artificial population data generation methodology. Static and dynamic population data are generated using the proved methodology. Static population represents the set V of units v_i, and dynamic population represents the set U of elements $u_{i,w}$. Artificial population data are necessary to carry out the simulation experiments needed in this research. The artificial population data are created from the data of the Statistical Household Register (SHR) and the survey data of the LFS. Random imputation techniques, survey data analysis and Markov Chain modelling are used to provide the results of the chapter.

2.1 Static Population

2.1.1 Statistical Household Register data

The data from SHR are given on the 30th January 2011. The list of residents living in private households is extracted from SHR using the standard procedures for creation of population frame.

The following procedures are applied to the data:

- Individuals aged 15–74 on the 30th January 2011 are selected. Other individuals are deleted.
- Individuals declared in dwellings with 12 or more individuals (aged 15–74 on the 30th January 2011) are deleted.

The result is a data frame with 1 705 048 records (individuals) and variables described in Table 2.1. These variables will be auxiliary information variables used in a simulation.

2.1.2 Labour Force Survey Data

LFS data are used to create study variables for the artificial population. The LFS data from the period 2007–2010 are used. The data describing individuals aged 15–74 are used. The result is a data frame with variables described in Table 2.2.

Table 2.1

SHR Variables

Variable	Description
iec2010	The number of census counting area
ATVK	The code of administrative territory
ind_maja	The ID of building
ind_dziv	The ID of flat (if there are flats in a building)
MAJOKLIS	The artificial ID of dwelling (if necessary)
coord_x	Geographical coordinate (X) of a building
coord_y	Geographical coordinate (Y) of a building
PERSKODS	Individualal ID
DZIMUMS	Sex
dzdat	Date of birth
vec	Age (on the 30th January 2011)

Table 2.2

LFS Variables

Variable	Description
apsek	Survey Period (year and quarter)
FPrimary	Household ID
Instance	Sequence number of individual in the household
ATVK	The code of administrative territory
B11	Sex
vec	Age (on Sunday of a reference week)
J100	Registered unemployment status
eka	Economic activity status by LFS methodology
E59	Number of hours per week usually worked in the main job

2.1.3 Data Merge

The units in SHR data are individuals and dwellings. The units in LFS data are individuals and households. The relation between units is:

1. There is one or more households in a dwelling.
2. There is one or more individuals in a household.

There is an assumption – the dwellings in SHR data correspond to the households in LFS data. There is a single household in each dwelling in other words. *The assumption is close to reality because there is not many dwellings with several households in Latvia.* The dwellings in SHR data will be called households hereinafter.

The aim of the data merge is to assign information from the LFS data to all households in the SHR data. This data merge can be considered as a data imputation where recipients are the households in the SHR data and donors are the households in the LFS data (United Nations, 2010). 53 variables were created to perform the imputation (see Table 2.3).

Imputation is done in seven levels. Households are imputed at the first five levels. Imputation units (households or individuals) are grouped by imputation groups as shown in Table 2.4.

There are G imputation groups in the imputation level k. The population of (households or individuals) from SHR is denoted by U. The population of (households or individuals) from LFS is denoted by V. U and V is split in G imputation groups. The groups are numbered sequentially – $1, \ldots, g, \ldots, G$. Both populations are split in subsets by the imputation groups – U_g and V_g where g is an index of the groups – $g = 1, \ldots, G$. The units from the populations U and V are denoted by u_i and v_j accordingly. The steps done for each unit (household or individual) in each imputation level:

1. Compute $|V_g|$ – the number of units in the group V_g (in LFS population).
2. If $|V_g| < 10$, a unit u_i is not imputed in the imputation level k.
3. If $|V_g| >= 10$, a unit v_j is randomly selected from the subset V_g and attached to the unit u_i. Data from unit v_j is imputed to the unit u_i.

The number of units imputed in each level is shown in Table 2.5.

The resulting file contains 1 705 264 records and 12 variables shown in Table 2.6.

2.2 Dynamic Population

The next task is to generate a dynamic population according to the description in Section 1.1. A variable – `eka` (economic activity status by LFS methodology) is extrapolated from the static population to the dynamic population. There are several assumptions incorporated in the population model:

- the set of units is fixed over W weeks,

Table 2.3

Variables defining the imputation groups

Variable	Values	Description
GR_A01	0, 1	Male in age group 15-19
GR_A02	0, 1	Male in age group 20-24
		…
GR_A11	0, 1	Male in age group 65-69
GR_A12	0, 1	Male in age group 70-74
GR_A13	0, 1	Female in age group 15-19
GR_A14	0, 1	Female in age group 20-24
		…
GR_A23	0, 1	Female in age group 65-69
GR_A24	0, 1	Female in age group 70-74
GR_B01	0, 1	Male in age group 15-19
GR_B02	0, 1	Male in age group 20-24
GR_B03	0, 1	Male in age group 25-34
GR_B04	0, 1	Male in age group 35-44
GR_B05	0, 1	Male in age group 45-54
GR_B06	0, 1	Male in age group 55-64
GR_B07	0, 1	Male in age group 65-74
GR_B13	0, 1	Female in age group 15-19
GR_B14	0, 1	Female in age group 20-24
GR_B15	0, 1	Female in age group 25-34
GR_B16	0, 1	Female in age group 35-44
GR_B17	0, 1	Female in age group 45-54
GR_B18	0, 1	Female in age group 55-64
GR_B19	0, 1	Female in age group 65-74
GR_C01	0, 1	Male in age group 15-19
GR_C02	0, 1	Male in age group 20-24
GR_C03	0, 1	Male in age group 25-44
GR_C04	0, 1	Male in age group 45-64
GR_C05	0, 1	Male in age group 65-74
GR_C13	0, 1	Female in age group 15-19
GR_C14	0, 1	Female in age group 20-24
GR_C15	0, 1	Female in age group 25-44
GR_C16	0, 1	Female in age group 45-64
GR_C17	0, 1	Female in age group 65-74
Strata	1, 2, 3, 4	Stratum
Reg	1, 2, 3, 4, 5, 6	Region
Dzimums	1, 2	Sex
Vecgr	1, 2, …, 12	Age group (five-year intervals)
Vec	15–74	Age

Table 2.4

Imputation levels

Level (k)	Imputation Unit	Variables Used
1	Household	GR_A01–GR_A24, Strata, Reg
2	Household	GR_B01–GR_B19, Strata, Reg
3	Household	GR_C01–GR_C17, Strata, Reg
4	Household	GR_C01–GR_C17, Strata
5	Household	GR_C01–GR_C17
6	Individual	Strata, Reg, Dzimums, vecgr, vec
7	Individual	Strata, Reg, Dzimums, vecgr

Table 2.5

SHR population split by the imputation levels

Level (k)	Households		Individuals	
	Count	Proportion (%)	Count	Proportion (%)
1	413 799	54.1	583 884	34.2
2	90 696	11.9	227 085	13.3
3	72 136	9.4	201 353	11.8
4	68 069	8.9	206 228	12.1
5	56 609	7.4	182 443	10.7
6	NA	NA	303 438	17.8
7	NA	NA	617	0.0
Total	764 946	100.0	1 705 048	100.0

Table 2.6

Variables of the resulting data file

Variable	Description
H_ID	Household ID
P_ID	Sequence number of individual in the household
iec2010	The number of census counting area
reg	Region code
nov	County code
DZIMUMS	Sex
vec	Age (on the 30th January 2011)
eka	Economic activity status by LFS methodology
E59	Number of hours per week usually worked in the main job
J100	The status of registered unemployment
coord_x_p	Geographical coordinate (X) of a building (with random noise)
coord_y_p	Geographical coordinate (Y) of a building (with random noise)

- the background variables such as age and place of residence are fixed for all units during the W weeks,
- the study variables (for example employment status) attached to elements can change from week to week,
- the membership of individuals to households is fixed over W weeks.

2.2.1 Theoretical Model

Let y_i be the value of the economic activity status for the ith individual. Markov chain model (Carkova, 2001) is used to generate the dynamic population. Variable y_i can take three different values: $y_i \in \{1, 2, 3\}$:

- $y_i = 1$ if individual i is employed,
- $y_i = 2$ if individual i is unemployed,
- $y_i = 3$ if individual i is inactive.

The value of y_i is defined once in a week (on Sunday by LFS methodology). Let $y_{i,w}$ be the value of the economic activity status for the ith individual on the wth week. Let $y_{i,w}$ be random variables and the sequence $y_{i,0}, y_{i,1}, y_{i,2}, \ldots$ be a time-inhomogeneous Markov chain. The state space for the Markov chain is $\{1, 2, 3\}$. The probability of going from state k to state l in a week is

$$p_{i,k,l,w,w+1} = P\left(y_{i,w+1} = l | y_{i,w} = k\right),$$

and let the time-dependent transition matrix be

$$\boldsymbol{P}_{i,w,w+1} = \boldsymbol{P}_{w,w+1} = \begin{pmatrix} p_{1,1,w,w+1} & p_{1,2,w,w+1} & p_{1,3,w,w+1} \\ p_{2,1,w,w+1} & p_{2,2,w,w+1} & p_{2,3,w,w+1} \\ p_{3,1,w,w+1} & p_{3,2,w,w+1} & p_{3,3,w,w+1} \end{pmatrix}.$$

2.2.2 Estimation of Transition Matrices

The LFS data are used to estimate $\boldsymbol{P}_{w,w+13} = \boldsymbol{P}_{q,q+1}$ (q is an index of quarter). The feature of rotating panel of LFS (see Section 1.3.1 for more details about the LFS rotation) is used in the estimation. It is possible to estimate the transition matrix of Markov chain after 13 weeks $\boldsymbol{P}_{w,w+13}$ because of the rotation period is a quarter or 13 weeks.

The LFS data collected during the period from the 1st quarter of 2007 till the 4th quarter of 2011 (*2007Q1–2011Q4*) are used in the estimation. Four variables extracted from the LFS database are described in Table 2.7.

The LFS data covers 20 consecutive quarters. Two indexes are used for quarters. See Table 2.8 for details:

Table 2.7

Variables from the LFS data used for the estimation of the transition matrix

Variable	Description
apsek	Survey Period (year and quarter)
FPrimary	Household ID
b06_ip	Individualal ID – traceable over waves
eka	Economic activity status by LFS methodology

Table 2.8

Indexes q and Q

Period	q	Q
20071	1	1
20072	2	2
20073	3	3
20074	4	4
20081	5	1
20082	6	2
20083	7	3
20084	8	4
...		
20111	17	1
20112	18	2
20113	19	3
20114	20	4

- $q \in \overline{1,20}$ – index referring to 20 consecutive quarters (there are 20 quarters in the LFS data used),
- $Q \in \overline{1,4}$ – index referring to 4 seasonal quarters (there are 4 quarters in a year).

It is possible to observe 19 pairs of consecutive quarters. See Table 2.9 for an illustration.

The $\boldsymbol{P}_{w,w+13}$ is estimated from each pair of quarters. The set of respondents is defined for each pair of quarters. The set of respondents for quarters q and $q+1$ is defined as respondents who responded in both quarters – q and $q+1$. Firstly the frequency matrix for each pair of quarters is computed as

$$\boldsymbol{A}_{q,q+1} = \begin{pmatrix} \sum_i (y_{i,q,q+1} = \{1,1\}) & \sum_i (y_{i,q,q+1} = \{1,2\}) & \sum_i (y_{i,q,q+1} = \{1,3\}) \\ \sum_i (y_{i,q,q+1} = \{2,1\}) & \sum_i (y_{i,q,q+1} = \{2,2\}) & \sum_i (y_{i,q,q+1} = \{2,3\}) \\ \sum_i (y_{i,q,q+1} = \{3,1\}) & \sum_i (y_{i,q,q+1} = \{3,2\}) & \sum_i (y_{i,q,q+1} = \{3,3\}) \end{pmatrix}$$

where $\sum_i (y_{i,q,q+1} = \{1,1\})$ is the count of respondents who were employed in quarter q ($y_{i,q} = 1$) and where employed also in quarter $q+1$ ($y_{i,q+1} = 1$), $\sum_i (y_{i,q,q+1} = \{1,2\})$ is the count of respondents who where employed in quarter q ($y_{i,q} = 1$) and where unemployed in quarter $q+1$

Table 2.9

Pairs of quarters

Period 1	Period 2	$Q1:Q2$	$Q2:Q3$	$Q3:Q4$	$Q4:Q1$
20071	20072	✓	·	·	·
20072	20073	·	✓	·	·
20073	20074	·	·	✓	·
20074	20081	·	·	·	✓
	...				
20101	20102	✓	·	·	·
20102	20103	·	✓	·	·
20103	20104	·	·	✓	·
20104	20111	·	·	·	✓
20111	20112	✓	·	·	·
20112	20113	·	✓	·	·
20113	20114	·	·	✓	·

($y_{i,q+1} = 2$) etc. $\boldsymbol{P}_{q,q+1}$ for each pair of quarters is estimated as

$$\hat{\boldsymbol{P}}_{q,q+1} = \begin{pmatrix} \frac{\sum_i(y_{i,q,q+1}=\{1,1\})}{\sum_i(y_{i,q}=1)} & \frac{\sum_i(y_{i,q,q+1}=\{1,2\})}{\sum_i(y_{i,q}=1)} & \frac{\sum_i(y_{i,q,q+1}=\{1,3\})}{\sum_i(y_{i,q}=1)} \\ \frac{\sum_i(y_{i,q,q+1}=\{2,1\})}{\sum_i(y_{i,q}=2)} & \frac{\sum_i(y_{i,q,q+1}=\{2,2\})}{\sum_i(y_{i,q}=2)} & \frac{\sum_i(y_{i,q,q+1}=\{2,3\})}{\sum_i(y_{i,q}=2)} \\ \frac{\sum_i(y_{i,q,q+1}=\{3,1\})}{\sum_i(y_{i,q}=3)} & \frac{\sum_i(y_{i,q,q+1}=\{3,2\})}{\sum_i(y_{i,q}=3)} & \frac{\sum_i(y_{i,q,q+1}=\{3,3\})}{\sum_i(y_{i,q}=3)} \end{pmatrix}$$

There are five estimates of transition matrix available for the each pair of seasonal quarters – $Q1:Q2$, $Q2:Q3$ and $Q3:Q4$. There are four estimates of transition matrix available for the pair $Q4:Q1$ (see Table 2.9). The estimates of transition matrix for the pairs of seasonal quarters are computed as average of transition matrices from according pairs of quarters.

$$\hat{\boldsymbol{P}}_{Q,Q+1} = \frac{\sum\limits_{((q-1) \bmod 4)+1=Q \ \& \ q \in \overline{1,19}} \hat{\boldsymbol{P}}_{q,q+1}}{\sum(((q-1) \bmod 4) + 1 = Q \ \& \ q \in \overline{1,19})}$$

Please note

$$Q + 1 := (Q \bmod 4) + 1 = \begin{cases} Q+1, & \text{if } \ Q < 4, \\ 1, & \text{if } \ Q = 4. \end{cases}$$

The estimate of $\boldsymbol{P}_{w,w+1}$ is computed as 13th root of the estimates of quarterly transition matrices

$$\hat{\boldsymbol{P}}_{w,w+1} = \begin{cases} \sqrt[13]{\hat{\boldsymbol{P}}_{1,2}} & \text{if } ((w-1) \bmod 52) + 1 \in \overline{1,13} \\ \sqrt[13]{\hat{\boldsymbol{P}}_{2,3}} & \text{if } ((w-1) \bmod 52) + 1 \in \overline{14,26} \\ \sqrt[13]{\hat{\boldsymbol{P}}_{3,4}} & \text{if } ((w-1) \bmod 52) + 1 \in \overline{27,39} \\ \sqrt[13]{\hat{\boldsymbol{P}}_{4,1}} & \text{if } ((w-1) \bmod 52) + 1 \in \overline{40,52} \end{cases}$$

Table 2.10

The Estimates of Transition matrices and their stationary distributions

Q	w	$\hat{\boldsymbol{P}}_{Q,Q+1}$	$\hat{\boldsymbol{P}}_{w,w+1}$	$\hat{\boldsymbol{\pi}}$
1	$\overline{1,13}$	$\begin{pmatrix} 0.950 & 0.021 & 0.029 \\ 0.251 & 0.541 & 0.209 \\ 0.058 & 0.052 & 0.890 \end{pmatrix}$	$\begin{pmatrix} 0.996 & 0.002 & 0.002 \\ 0.025 & 0.952 & 0.022 \\ 0.004 & 0.006 & 0.990 \end{pmatrix}$	$\begin{pmatrix} 0.649 \\ 0.063 \\ 0.289 \end{pmatrix}$
2	$\overline{14,26}$	$\begin{pmatrix} 0.944 & 0.021 & 0.035 \\ 0.253 & 0.540 & 0.206 \\ 0.055 & 0.055 & 0.891 \end{pmatrix}$	$\begin{pmatrix} 0.995 & 0.002 & 0.003 \\ 0.026 & 0.952 & 0.022 \\ 0.004 & 0.006 & 0.990 \end{pmatrix}$	$\begin{pmatrix} 0.612 \\ 0.066 \\ 0.321 \end{pmatrix}$
3	$\overline{27,39}$	$\begin{pmatrix} 0.937 & 0.028 & 0.035 \\ 0.199 & 0.609 & 0.192 \\ 0.048 & 0.042 & 0.910 \end{pmatrix}$	$\begin{pmatrix} 0.995 & 0.003 & 0.003 \\ 0.019 & 0.962 & 0.019 \\ 0.004 & 0.004 & 0.992 \end{pmatrix}$	$\begin{pmatrix} 0.541 \\ 0.080 \\ 0.379 \end{pmatrix}$
4	$\overline{40,52}$	$\begin{pmatrix} 0.930 & 0.033 & 0.037 \\ 0.183 & 0.596 & 0.221 \\ 0.042 & 0.043 & 0.915 \end{pmatrix}$	$\begin{pmatrix} 0.994 & 0.003 & 0.003 \\ 0.018 & 0.960 & 0.022 \\ 0.003 & 0.004 & 0.993 \end{pmatrix}$	$\begin{pmatrix} 0.482 \\ 0.085 \\ 0.433 \end{pmatrix}$

13th root of $\hat{\boldsymbol{P}}_{1,2}$ is computed by the help of eigen decomposition of the matrix. The eigenvalues and eigenvectors of the matrix are computed by the R function `eigen` (R Core Team, 2013). 13th root of $\hat{\boldsymbol{P}}_{1,2}$ is computed as

$$\sqrt[13]{\hat{\boldsymbol{P}}_{Q,Q+1}} = \boldsymbol{A}\sqrt[13]{\boldsymbol{D}}\boldsymbol{A}^{-1}$$

where $\boldsymbol{A}$ is the matrix of eigenvectors of the matrix $\hat{\boldsymbol{P}}_{Q,Q+1}$ and $\boldsymbol{D}$ is diagonal matrix with the eigenvalues of the matrix $\hat{\boldsymbol{P}}_{Q,Q+1}$ on the diagonal. The estimate of stationary distribution $\hat{\boldsymbol{\pi}}$ is computed as

$$\hat{\boldsymbol{\pi}} = \text{diag}\left(\lim_{k\to\infty} \hat{\boldsymbol{P}}^{k}_{w,w+1}\right)$$

2.2.3 Generation of Dynamic Population

The dynamic population is generated using the estimated transition matrices (Table 2.10). The population is generated for 312 weeks (six years). The resulting distribution of variable `eka` is displayed in Figure 2.1.

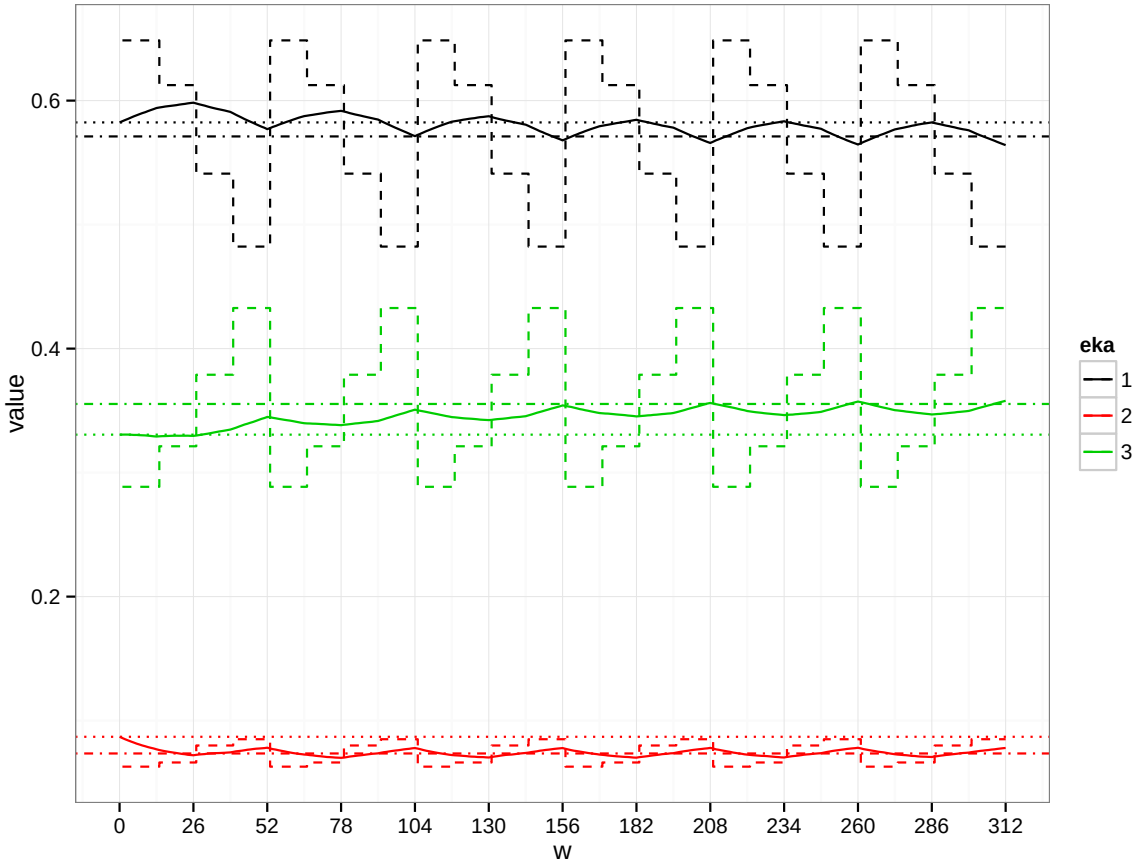


Figure 2.1 **The distribution of `eka` in dynamic population. Colours represent three states of `eka`. Solid line – weekly proportion, dashed line – quarterly stationary distribution, dotted line – initial state (week 0) from the static population, dot-dashed line – the average stationary distribution.**

Chapter 3

Cost Efficiency – Theoretical Model

Definition of design cost efficiency is introduced in the chapter. Modified stratified simple random sampling design is defined and the expression of variance for population parameter estimates (total and ratio of two totals) under the design is given. The estimation formulae of the Monte Carlo simulation experiment are provided. Theory of survey sampling, mathematical statistics, linearisation techniques and Monte Carlo simulation experiments is used to provide the results of the chapter.

3.1 Definition of Cost Efficiency

Assume an arbitrary population parameter θ. Assume there is a probability sample s drawn by sampling design $p(s)$. θ can be estimated using an estimator $\hat{\theta}_p$. The variance of $\hat{\theta}_p$ is denoted by $\text{Var}\left(\hat{\theta}_p\right)$. There is a cost function $c(s)$. The cost of sample s can be computed by the cost function $c_s = c(s)$. The cost c_s is a random variable because s is a random sample. The expectation of c_s under sampling design $p(s)$ is denoted as $\text{E}(c_s) = C_p$.

Usual desire is to minimise $\text{Var}\left(\hat{\theta}_p\right)$ and C_p. Unfortunately these are conflicting tasks. One has to increase cost to reduce variance and variance goes up when cost is reduced. The task of a statistician is to find sampling design $p(s)$ so that C_s and $\text{Var}\left(\hat{\theta}_p\right)$ would be in "balance". There is a need for a measure of cost efficiency. Assume two sampling designs:

- Simple random sampling – srs
- Alternative sampling design – $p(s)$

The classical design effect for sampling design $p(s)$ by Kish (1965) is the ratio of variances under condition of equal sample sizes defined by

$$\text{deff}\left(p, \hat{\theta}_p, n\right) = \frac{\text{Var}_p\left(\hat{\theta}_p \middle| \text{E}(n_p) = n\right)}{\text{Var}_{srs}\left(\hat{\theta}_{srs} \middle| n_{srs} = n\right)},$$

where $\hat{\theta}_p$ denotes estimator of θ under sampling design $p(s)$, $\hat{\theta}_{srs}$ denotes estimator of θ under simple random sampling. Alternative design effect is introduced by

$$\text{deff}^{\star}\left(p, \hat{\theta}_p, \gamma\right) = \frac{\text{Var}_p\left(\hat{\theta}_p \middle| C_p \approx \gamma\right)}{\text{Var}_{srs}\left(\hat{\theta}_{srs} \middle| C_{srs} \approx \gamma\right)}$$

where γ is a survey budget available for survey fieldwork. It is defined as a ratio of variances under condition of equal expected cost. Approximate equality ($\approx$) is used in the definition of alternative design effect because γ can take any positive value, but the values of C_p and C_{srs} can take only values from a finite subset of $\mathbb{R}^+$. The alternative design effect could be used as a measure of cost efficiency. Assume two sampling designs – $p(s)$ and $q(s)$.

Definition 1. The sampling design $p(s)$ is more cost efficient than the sampling design $q(s)$ for estimation of θ with survey budget γ if $\text{deff}^{\star}\left(p, \hat{\theta}_p, \gamma\right) < \text{deff}^{\star}\left(q, \hat{\theta}_q, \gamma\right)$.

Definition 1 is equivalent to Definition 2.

Definition 2. The sampling design $p(s)$ is more cost efficient than the sampling design $q(s)$ for estimation of θ with survey budget γ if $\text{Var}_p\left(\hat{\theta}_p \middle| C_p \approx \gamma\right) < \text{Var}_q\left(\hat{\theta}_q \middle| C_q \approx \gamma\right)$.

3.2 Aim of the Research

The aims of this chapter and the next chapter are to:

- describe the methodology for comparing sampling designs regarding the cost efficiency,
- compare several sampling designs with the developed methodology.

Definition 2 is used to compare the sampling designs. The aims are achieved in several steps.

1. Selection of population parameters for the analysis.
2. Definition of the cost function $c_s = c(s)$.
3. Selection of sampling designs and definition of the estimators for the population parameters selected at the step 1.
4. Calculation of sample size for each chosen sampling design to achieve approximately equal expected cost for all designs.
5. Calculation of variance for the estimates of population parameters selected at the step 1.
6. Determination of the most cost efficient sampling design.

3.3 Sampling Designs

The selection of sampling designs is restricted by the requirements of LFS:

- LFS is a continuous survey. The population is observed weekly. The sampling units have to be allocated evenly by weeks (European Commission, 2012b, p. 5).

- Two questionnaires are used – a household questionnaire and an individual questionnaire.
- The response burden on respondents has to be minimised if possible.

3.3.1 mSRS Design

Modified simple random sampling design (denoted as mSRS further) is introduced. The target population is described in Section 1.1. The classical simple random sample (SRS) by selecting $n \leq N$ elements from the population U would violate two requirements of the LFS:

- sample would not be evenly distributed by weeks for most of realised samples by SRS,
- it would be possible to select more than one element corresponding to the same unit during W weeks (observe a unit (individual or household) more than once during W weeks).

These are the reasons for introduction of the mSRS sampling design.

New notation is introduced. Let $\tilde{s}$ denote the set of sampled units, $\tilde{s} \subseteq V$, s_w denote the set of sampled elements in the week w, $s_w \subseteq U_w$, and s denote the set of sampled elements over W weeks, $s = \cup_{w=1}^{W} s_w \subseteq U$.

The weekly sample size m is chosen. Total sample size n is computed as mW. The value of m has to be chosen so that $n = mW \leq M$, because each unit can be sampled only once during W weeks.

The goals of the mSRS sampling design are:

- all elements of U have sampling probabilities equal to $\pi_k = \frac{n}{N} = \frac{m}{M}$,
- all weekly samples are realised with equal sample size $\forall w : |s_w| = m$,
- total sample size is equal to n, $n = |s| = Wm$,
- all n sampled elements refer to n different units (one and only one element $u_{i,w}$ may be sampled for the unit v_i).

There are several techniques to achieve the sample by mSRS design. An example is presented here. The sample is selected in two steps, where n units are selected by simple random sampling without replacement from M units and sorted in the random order at the first step. The ordered sample of units is systemically split in W blocks with length m at the second step. The units of the first block determine sampled elements for the first week. The units of the second block determine sampled elements for the second week. The procedure is continued till finally the units of the last block determine sampled elements for the last week.

For example, assume $M = 100$, $W = 3$, $N = MW = 300$, $m = 5$, $n = mW = 15 \leq 100$, $i \in \overline{1, M}$ is the index of units. Then

- the realisation of randomly ordered list of sampled units is:

$$\tilde{s} = \{v_{30}, v_8, v_{92}, v_{68}, v_{80}, v_{91}, v_{37}, v_{75}, v_9, v_{13}, v_{39}, v_{29}, v_{33}, v_{58}, v_{76}\},$$

- the realised weekly element samples are

$$s_1 = \left\{u_{8,1}, u_{30,1}, u_{68,1}, u_{80,1}, u_{92,1}\right\},$$
$$s_2 = \left\{u_{9,2}, u_{13,2}, u_{37,2}, u_{75,2}, u_{91,2}\right\},$$
$$s_3 = \left\{u_{29,3}, u_{33,3}, u_{39,3}, u_{58,3}, u_{76,3}\right\},$$

- total realised element sample with indices $k \in \overline{1, N}$ is

$$s = \left\{u_8, u_{30}, u_{68}, u_{80}, u_{92}, u_{109}, u_{113}, u_{137}, u_{175}, u_{191}, u_{229}, u_{233}, u_{239}, u_{258}, u_{276}\right\}.$$

Probability to select the unit i in the sample of units at the first step is $\frac{n}{M}$. The probability of unit i to be located in the block w after random ordering is $\frac{1}{W}$. The sampled element is determined by the index i of the sampled unit and the index w of the block containing the unit i. Therefore the sampling probability of any element is equal to $\pi_{i,w} = \pi_k = \frac{n}{M} \cdot \frac{1}{W} = \frac{m}{M}$.

3.3.2 mSSRS Design

Stratified mSRS is realised if the population units are stratified in H strata and mSRS is applied independently in each stratum. This design is denoted mSSRS.

The population of units V is stratified in non-overlapping strata $V_1, V_2, \ldots, V_H$ where H is the number of strata, $\cup_{h=1}^{H} V_h = V$. The number of units in the population of each stratum is denoted by M_h, $\sum_{h=1}^{H} M_h = M$. The number of weeks W is chosen and weekly sample size m_h for each stratum is set. Then mSRS is applied independently in each stratum. The sampling probabilities for elements depending on stratum h are equal to $\pi_{h,i,w} = \pi_{h,k} = \frac{m_h}{M_h}$.

3.3.3 Choice of Sampling Designs for the Cost Efficiency Study

Three sampling designs are chosen for the cost efficiency study.

1. mSSRS with individuals as sampling units (denoted mSSRSi). Each sampled individual is interviewed by the individual questionnaire and also the household questionnaire.
2. mSSRS with households as sampling units (denoted mSSRSh). Each sampled household is interviewed by the household questionnaire and all household members are interviewed by the individual questionnaires.
3. Stratified systematic two-stage sampling design used in practice for Latvian LFS and described in Chapter 1 (denoted TSSh). Households are sampling units for this design. Each sampled household is interviewed by the household questionnaire and all household members are interviewed by the individual questionnaires.

It is possible to draw conclusions about the expected precision of the designs under the assumption of equal expected sample size with analytically based considerations. Sample size is defined as expected number of sampled individuals in this example. Assume the same stratifica-

tion for all designs and only individual questionnaires are used in survey. The expected interview cost is approximately equal for all designs because of equal expected sample sizes (number of individual questionnaires to be filled). The expected travel cost is highest for the mSSRSi (sample is well spread and a single individual is interviewed from each household in most cases), lower for the mSSRSh (sample is well spread but two individuals are interviewed from each household on average) and the lowest for the TSSh (sample is geographically clustered and two individuals are interviewed from each household on average). The expected fieldwork cost is the sum of expected interview cost and the expected travel cost – so the expected fieldwork cost is highest for the mSSRSi, followed by the mSSRSh and the lowest expected fieldwork cost is for the TSSh.

The precision of population parameter estimates is driven mainly by two effects – stratification and clustering. Both effects can increase or decrease the variance of population parameter estimate depending on the population under study and sampling design used. Equal stratification is assumed in this example. Clustering of units is the technique used in sampling when population units are grouped in clusters and clusters are used as sampling units. Clustering is not used by mSSRSi. The clustering of individuals in households is used by mSSRSh. The clustering of individuals in households and clustering of households in census counting areas are used in TSSh design. Some calculations have shown that units tend to be more homogeneous in clusters (households and census counting areas) compared to whole population in the case of Latvian LFS. The level of homogeneity of units in clusters is described by the coefficient of intraclass correlation. Positive coefficient of intraclass correlation is the sign that clustering of population units in sample can result with higher variance of population parameter estimates compared to non-clustered sampling design. It is possible to conclude that clustering tends to increase the variance of population parameter estimates in the case of Latvian LFS. The effect of increasing variance of estimates due to clustering is called cluster effect. See Kish (1965, p. 161) for more details regarding clustering and the coefficient of intraclass correlation.

The cluster effect will be highest for TSSh (because of geographical clustering and clustering of individuals by households), lower for mSSRSh (only clustering of individuals by households) and the lowest for mSSRSi (no clustering of individuals). The expected precision in this case is reverse to cluster effect (because of equal sample sizes). Therefore it is possible to draw a rough conclusion that mSSRSi would be the most precise design followed by mSSRSh and TSSh under the assumption of equal expected sample size.

See Table 3.1 for illustration. The stars denote the order of sampling designs by a chosen variable, for example, travel cost is the lowest for the TSSH (*), it is higher for the mSSRSh (**) and the highest for the mSSRSi (***). Equal number of stars means that order can not be determined.

The situation is different if the expected fieldwork cost is set to be approximately equal for all designs. The expectation is that travel cost will be highest for mSSRSi followed by mSSRSh and TSSh. Interview cost will be lowest for mSSRSi to compensate the highest travel cost,

Table 3.1

Rough precision evaluation under assumption of equal sample size

Design	TravCost	IntCost	FWCost	SamplSize	ClustEff	Precision
mSSRSi	***	**	***	**	*	***
mSSRSh	**	**	**	**	**	**
TSSh	*	**	*	**	***	*

Table 3.2

Rough precision evaluation under assumption of equal fieldwork cost

Design	TravCost	IntCost	FWCost	SamplSize	ClustEff	Precision
mSSRSi	***	*	**	*	*	?
mSSRSh	**	**	**	**	**	?
TSSh	*	***	**	***	***	?

the second highest interview cost will be for mSSRSh and the highest for TSSh. Travel cost and interview cost will be chosen so to equalise the expected fieldwork cost for all designs. The equalised fieldwork cost will result with the lowest sample size for mSSRSi, higher for mSSRSh and the highest for TSSh. The designs will have the same cluster effect as described before – the highest for TSSh, followed by mSSRSh and mSSRSi. It is not possible to give an evaluation about the precision in this case without an additional in-depth analysis. mSSRSi has the lowest cluster effect but at the same time also the lowest sample size. On contrary TSSh has the highest sample size but also the highest cluster effect. There is no obvious answer to the question which of the chosen sampling design is the most precise under the assumption of approximately equal fieldwork cost (see Table 3.2).

3.4 Estimators of Population Parameters

The sample of individuals is generated by all three designs and it is possible to compute the sampling probabilities of individuals for each design. The estimators for Y_q and R_q (see the equations 1.1 and 1.2) are constructed using π estimator (Särndal et al., 1992, p. 42, 176):

$$\hat{Y}_q = \frac{1}{13} \sum_{(i,w)\in s} \frac{y_{i,w}}{\pi_{i,w}}, \tag{3.1}$$

$$\hat{R}_q = \frac{\hat{Y}_q}{\hat{Z}_q} = \frac{\sum_{(i,w)\in s} \frac{y_{i,w}}{\pi_{i,w}}}{\sum_{(i,w)\in s} \frac{z_{i,w}}{\pi_{i,w}}}. \tag{3.2}$$

where s is a probability sample of observations from U, $s \subseteq U$, $y_{i,w}$ and $z_{i,w}$ are values assigned to observation $u_{i,w}$, and $\pi_{i,w}$ is the probability of observation $u_{i,w}$ to be included in sample s.

3.5 Variance under mSRS Sampling Design

3.5.1 Variance for Estimator of Total

The target population is described in Section 1.1 and the mSRS design is described in Section 3.3. Some notation is repeated here for clarity purposes. Population V is a finite set of units. The units of V are individuals (there are cases when households are used as units). The size of V is M. A weekly sample size is denoted by m. The values of M and m are constant over time. The index i is used to label units, $i \in \overline{1, M}$. The total of variable y in week w is defined as

$$Y_w = \sum_{i=1}^{M} y_{i,w}. \tag{3.3}$$

The number of weeks observed is denoted by W. The index w is used to label weeks, $w \in \overline{1, W}$. The number of elements in population U is denoted by $N = WM$ (see Table 1.1 as an example). The sample size of elements is denoted by $n = mW$. The index k is used to label elements, $k \in \overline{1, N}$. The value of m has to be chosen so that $n = mW \leq M$, because each unit can be sampled only once during W weeks.

The column vector of values y_k is denoted by $\boldsymbol{y} = (y_1, y_2, \ldots, y_k, \ldots, y_N)'$. Alternatively $\boldsymbol{y} = (\boldsymbol{y}'_1, \boldsymbol{y}'_2, \ldots, \boldsymbol{y}'_w, \ldots, \boldsymbol{y}'_W)'$, where $\boldsymbol{y}_w$ is a column vector of values $y_{i,w}$ at the week w. The total of $\boldsymbol{y}$ over W weeks is defined as[1]

$$Y = \sum_w Y_w = \sum_w \sum_{i=1}^{M} y_{i,w} = \sum_{k=1}^{N} y_k,$$

and the π-estimator is given by

$$\hat{Y} = \sum_w \hat{Y}_w = \sum_w \sum_{i \in s} \frac{y_{i,w}}{\pi_{i,w}} = \sum_{k \in s} \frac{y_k}{\pi_k},$$

where s is a probability sample of elements, and $\pi_{i,w}$ (π_k) is inclusion probability of element $u_{i,w}$ (u_k) in a sample. The variance of the estimator is derived using the general variance expression from Särndal et al. (1992, p. 44) as

$$\text{Var}\left(\hat{Y}\right) = \sum_{k=1}^{N} \sum_{l=1}^{N} \frac{\pi_{kl}}{\pi_k \pi_l} y_k y_l - \left(\sum_{k=1}^{N} y_k\right)^2, \tag{3.4}$$

where π_{kl} is the probability that both elements u_k and u_l are simultaneously included in the sample, $\pi_{kl} = \pi_k \pi_{l,k \in s} = p\left(k \in s\right) p\left(l \in s | k \in s\right)$. Assume the order of elements each week equal to the order of units – indices $\{k : k \bmod M = i\}$ correspond to the unit v_i. Then $\pi_k =$

[1] The notation $\sum_w$ is used instead of $\sum_{w=1}^{W}$ further to simplify notation

$\pi = \frac{n}{N} = \frac{m}{M}$ for all k, but there are four possible values of π_{kl}:

$$\pi_{kl} = \begin{cases} \frac{m}{M} & \text{if } (k = l), \\ \frac{m}{M}\frac{m-1}{M-1} & \text{if } \left(\lfloor\frac{k-1}{M}\rfloor = \lfloor\frac{l-1}{M}\rfloor\right) \ \& \ (k \neq l), \\ 0 & \text{if } (k \bmod M = l \bmod M) \ \& \ (k \neq l), \\ \frac{m}{M}\frac{m}{M-1} & \text{if } \left(\lfloor\frac{k-1}{M}\rfloor \neq \lfloor\frac{l-1}{M}\rfloor\right) \ \& \ (k \bmod M \neq l \bmod M). \end{cases}$$

Some explanation of π_{kl}:

- $(k = l)$: elements u_k and u_l is the same element, so $p\left(l \in s | k \in s\right) = 1 \Rightarrow \pi_{kl} = \pi_k = \frac{m}{M}$.
- $\left(\lfloor\frac{k-1}{M}\rfloor = \lfloor\frac{l-1}{M}\rfloor\right) \ \& \ (k \neq l)$: elements u_k and u_l are sampled from the same week, but $u_k \neq u_l$. Assume element u_k is sampled in week w. There are $M-1$ other elements in the population of the week w and $m-1$ elements of them are sampled, so $p\left(l \in s | k \in s\right) = \frac{m-1}{M-1} \Rightarrow \pi_{kl} = \frac{m}{M}\frac{m-1}{M-1}$.
- $(k \bmod M = l \bmod M) \ \& \ (k \neq l)$: elements u_k and u_l refer to the same unit, but they are sampled in different weeks. The mSRS design does not allow a unit to be sampled more than once in a period of W weeks, so $p\left(l \in s | k \in s\right) = 0 \Rightarrow \pi_{kl} = 0$.
- $\left(\lfloor\frac{k-1}{M}\rfloor \neq \lfloor\frac{l-1}{M}\rfloor\right) \ \& \ (k \bmod M \neq l \bmod M)$: elements u_k and u_l are sampled from distinct weeks and u_k and u_l refer to different units. Assume element u_k is sampled in week w, so element u_l is sampled from any other week different from w, and element u_l can not refer to the same unit as element u_k. There are $N - M - (W-1)$ possible candidates for u_l and $n - m$ elements of them are sampled, so $p\left(l \in s | k \in s\right) = \frac{n-m}{N-M-(W-1)} \Rightarrow \pi_{kl} = \frac{m}{M}\frac{Wm-m}{WM-M-(W-1)} = \frac{m}{M}\frac{(W-1)m}{(W-1)(M-1)} = \frac{m}{M}\frac{m}{M-1}$.

The matrix of π_{kl} is denoted by $\mathbf{\Pi}$. $\mathbf{\Pi}$ is a square matrix with size $N \times N$. An example of $\mathbf{\Pi}$ for the parameters $M = 6$, $W = 3$, $N = 18$, $m = 2$ and $n = 6$ is given below.

$$\mathbf{\Pi} = \left(\begin{array}{cccccc:cccccc:cccccc}
\pi_k & \star & \star & \star & \star & \star & 0 & \cdot & \cdot & \cdot & \cdot & \cdot & 0 & \cdot & \cdot & \cdot & \cdot & \cdot \\
\star & \pi_k & \star & \star & \star & \star & \cdot & 0 & \cdot & \cdot & \cdot & \cdot & \cdot & 0 & \cdot & \cdot & \cdot & \cdot \\
\star & \star & \pi_k & \star & \star & \star & \cdot & \cdot & 0 & \cdot & \cdot & \cdot & \cdot & \cdot & 0 & \cdot & \cdot & \cdot \\
\star & \star & \star & \pi_k & \star & \star & \cdot & \cdot & \cdot & 0 & \cdot & \cdot & \cdot & \cdot & \cdot & 0 & \cdot & \cdot \\
\star & \star & \star & \star & \pi_k & \star & \cdot & \cdot & \cdot & \cdot & 0 & \cdot & \cdot & \cdot & \cdot & \cdot & 0 & \cdot \\
\star & \star & \star & \star & \star & \pi_k & \cdot & \cdot & \cdot & \cdot & \cdot & 0 & \cdot & \cdot & \cdot & \cdot & \cdot & 0 \\
\hdashline
0 & \cdot & \cdot & \cdot & \cdot & \cdot & \pi_k & \star & \star & \star & \star & \star & 0 & \cdot & \cdot & \cdot & \cdot & \cdot \\
\cdot & 0 & \cdot & \cdot & \cdot & \cdot & \star & \pi_k & \star & \star & \star & \star & \cdot & 0 & \cdot & \cdot & \cdot & \cdot \\
\cdot & \cdot & 0 & \cdot & \cdot & \cdot & \star & \star & \pi_k & \star & \star & \star & \cdot & \cdot & 0 & \cdot & \cdot & \cdot \\
\cdot & \cdot & \cdot & 0 & \cdot & \cdot & \star & \star & \star & \pi_k & \star & \star & \cdot & \cdot & \cdot & 0 & \cdot & \cdot \\
\cdot & \cdot & \cdot & \cdot & 0 & \cdot & \star & \star & \star & \star & \pi_k & \star & \cdot & \cdot & \cdot & \cdot & 0 & \cdot \\
\cdot & \cdot & \cdot & \cdot & \cdot & 0 & \star & \star & \star & \star & \star & \pi_k & \cdot & \cdot & \cdot & \cdot & \cdot & 0 \\
\hdashline
0 & \cdot & \cdot & \cdot & \cdot & \cdot & 0 & \cdot & \cdot & \cdot & \cdot & \cdot & \pi_k & \star & \star & \star & \star & \star \\
\cdot & 0 & \cdot & \cdot & \cdot & \cdot & \cdot & 0 & \cdot & \cdot & \cdot & \cdot & \star & \pi_k & \star & \star & \star & \star \\
\cdot & \cdot & 0 & \cdot & \cdot & \cdot & \cdot & \cdot & 0 & \cdot & \cdot & \cdot & \star & \star & \pi_k & \star & \star & \star \\
\cdot & \cdot & \cdot & 0 & \cdot & \cdot & \cdot & \cdot & \cdot & 0 & \cdot & \cdot & \star & \star & \star & \pi_k & \star & \star \\
\cdot & \cdot & \cdot & \cdot & 0 & \cdot & \cdot & \cdot & \cdot & \cdot & 0 & \cdot & \star & \star & \star & \star & \pi_k & \star \\
\cdot & \cdot & \cdot & \cdot & \cdot & 0 & \cdot & \cdot & \cdot & \cdot & \cdot & 0 & \star & \star & \star & \star & \star & \pi_k
\end{array}\right)$$

It is observable that:

- Matrix $\mathbf{\Pi}$ consists of W^2 square blocks of size $M \times M$. There are 9 blocks with size 6×6 in the example.
- Values of diagonal elements of $\mathbf{\Pi}$ are $\pi_{kl} = \pi_k = \frac{1}{3}$ in the example.
- Values of the cells marked by the star ($\star$) are $\pi_{kl} = \frac{m}{M}\frac{m-1}{M-1}$. Here π_{kl} is the probability to select two different units simultaneously in the sample of a week, $\pi_{kl} = \frac{1}{3}\frac{1}{5} = \frac{1}{15}$ in the example.
- Values of the diagonal elements of all non-diagonal blocks are $\pi_{kl} = 0$. Here π_{kl} is the probability to select a unit simultaneously in the samples of two different weeks. The design does not allow the unit to be sampled more than once during W weeks, so this probability is zero.
- Values of the cells marked by the dot ($\cdot$) are $\pi_{kl} = \frac{m}{M}\frac{m}{M-1}$. Here π_{kl} is the probability to select two different units simultaneously in the samples of two different weeks, $\pi_{kl} = \frac{1}{3}\frac{2}{5} = \frac{2}{15}$ in the example.

The matrix $\boldsymbol{Y} = \boldsymbol{y}\boldsymbol{y}'$ includes all cross-products $y_k y_l$. The structure of $\boldsymbol{Y}$ is the same as for $\boldsymbol{\Pi}$ – the size of $\boldsymbol{Y}$ is $N \times N$ and $\boldsymbol{Y}$ consists of W^2 square blocks each with size $M \times M$,

$$\boldsymbol{Y} = \begin{pmatrix} \boldsymbol{A}_{1,1} & \boldsymbol{B}_{1,2} & \boldsymbol{B}_{1,3} & \cdots & \boldsymbol{B}_{1,W} \\ \boldsymbol{B}_{2,1} & \boldsymbol{A}_{2,2} & \boldsymbol{B}_{2,3} & \cdots & \boldsymbol{B}_{2,W} \\ \boldsymbol{B}_{3,1} & \boldsymbol{B}_{3,2} & \boldsymbol{A}_{3,3} & \cdots & \boldsymbol{B}_{3,W} \\ & & \cdots & & \\ \boldsymbol{B}_{W,1} & \boldsymbol{B}_{W,2} & \boldsymbol{B}_{W,3} & \cdots & \boldsymbol{A}_{W,W} \end{pmatrix},$$

where $\boldsymbol{A}_{w,w} = \boldsymbol{y}_w \boldsymbol{y}_w'$ and $\boldsymbol{B}_{w,v} = \boldsymbol{y}_w \boldsymbol{y}_v'$.

New notation is introduced:

$$C = \sum_{w \neq v} \sum Y_w Y_v,$$

$$D = \sum_{w \neq v} \sum \sum_{i=1}^{M} y_{i,w} y_{v,i},$$

where Y_w and Y_v is defined by (3.3), C is the sum of the elements of the $\boldsymbol{B}_{w,v}$ matrices, and D is the sum of the diagonals of all $\boldsymbol{B}_{w,v}$ matrices.

$$\tilde{\boldsymbol{Y}} = \frac{1}{\pi^2} \left(\boldsymbol{\Pi} \odot \boldsymbol{Y} \right) = \begin{pmatrix} \tilde{\boldsymbol{A}}_{1,1} & \tilde{\boldsymbol{B}}_{1,2} & \tilde{\boldsymbol{B}}_{1,3} & \cdots & \tilde{\boldsymbol{B}}_{1,W} \\ \tilde{\boldsymbol{B}}_{2,1} & \tilde{\boldsymbol{A}}_{2,2} & \tilde{\boldsymbol{B}}_{2,3} & \cdots & \tilde{\boldsymbol{B}}_{2,W} \\ \tilde{\boldsymbol{B}}_{3,1} & \tilde{\boldsymbol{B}}_{3,2} & \tilde{\boldsymbol{A}}_{3,3} & \cdots & \tilde{\boldsymbol{B}}_{3,W} \\ & & \cdots & & \\ \tilde{\boldsymbol{B}}_{W,1} & \tilde{\boldsymbol{B}}_{W,2} & \tilde{\boldsymbol{B}}_{W,3} & \cdots & \tilde{\boldsymbol{A}}_{W,W} \end{pmatrix},$$

where $\pi = \frac{m}{M}$, symbol $\odot$ is a Hadamard product and $\tilde{\boldsymbol{Y}}$ is the $N \times N$ matrix.

The double sum in (3.4) is equal to the sum of the elements of $\tilde{\boldsymbol{Y}}$ matrix and it can be expressed by three addends as

$$\sum_{k=1}^{N} \sum_{l=1}^{N} \frac{\pi_{kl}}{\pi_k \pi_l} y_k y_l = \mathbf{1}' \tilde{\boldsymbol{Y}} \mathbf{1} = \left(\frac{M}{m} \right)^2 \mathbf{1}' \left(\boldsymbol{\Pi} \odot \boldsymbol{Y} \right) \mathbf{1} = \Gamma + \Delta + \Lambda,$$

where $\mathbf{1}$ is a column vector with length N and all entries equal to 1, and addends are

$$\Gamma = \sum_w \Gamma_w = \frac{M}{m} \sum_w \sum_{i=1}^{M} y_{i,w}^2,$$

$$\Delta = \sum_w \Delta_w = \frac{M}{m} \frac{m-1}{M-1} \sum_w \left(Y_w^2 - \sum_{i=1}^{M} y_{i,w}^2 \right),$$

$$\Lambda = \frac{M}{M-1} \left(C - D \right),$$

where Γ_w is the sum of the diagonal elements of $\tilde{\boldsymbol{A}}_{w,w}$, Δ_w is the sum of $\tilde{\boldsymbol{A}}_{w,w}$ elements outside diagonal, and Λ is the sum of all $\tilde{\boldsymbol{B}}_{w,v}$ elements. The last addend in (3.4) can be rewritten as

$$\left(\sum_{k=1}^{N} y_k\right)^2 = \left(\sum_{w} Y_w\right)^2 = \sum_{w} Y_w^2 + C,$$

and the alternative expression for (3.4) is

$$\text{Var}\left(\hat{Y}\right) = \Gamma + \Delta + \Lambda - C - \sum_{w} Y_w^2. \tag{3.5}$$

It is possible to derive for the week w:

$$\begin{aligned}
\Gamma_w + \Delta_w - Y_w^2 &= \frac{M}{m}\sum_{i=1}^{M} y_{i,w}^2 + \frac{M}{m}\frac{m-1}{M-1}\left(Y_w^2 - \sum_{i=1}^{M} y_{i,w}^2\right) - Y_w^2 = \\
&= \left(\frac{M}{m} - \frac{M}{m}\frac{m-1}{M-1}\right)\sum_{i=1}^{M} y_{i,w}^2 + \left(\frac{M}{m}\frac{m-1}{M-1} - 1\right)Y_w^2 = \\
&= \frac{M\,(M-m)}{m\,(M-1)}\sum_{i=1}^{M} y_{i,w}^2 + \frac{m-M}{m\,(M-1)}Y_w^2 = \\
&= \frac{M^2\left(1-\frac{m}{M}\right)}{m\,(M-1)}\sum_{i=1}^{M} y_{i,w}^2 - \frac{M^2\frac{1}{M}\left(1-\frac{m}{M}\right)}{m\,(M-1)}Y_w^2 = \\
&= M^2\frac{1-\frac{m}{M}}{m}\frac{1}{M-1}\left(\sum_{i=1}^{M} y_{i,w}^2 - \frac{1}{M}Y_w^2\right) = \\
&= M^2\frac{1-\frac{m}{M}}{m}S_w^2,
\end{aligned} \tag{3.6}$$

and it is equivalent to Var $\left(\hat{Y}_w\right)$. From (3.6) follows

$$\Gamma + \Delta - \sum_{w} Y_w^2 = M^2\frac{1-\frac{m}{M}}{m}\sum_{w} S_w^2. \tag{3.7}$$

The (3.7) is equal to $\sum_w$ Var $\left(\hat{Y}_w\right)$, and it is equal the variance of $\hat{Y}$ if stratified simple random sampling without replacement stratified by weeks with even sample allocation by weeks would

be used. There is a remainder from (3.5) and (3.4)

$$\begin{aligned}\text{Var}\left(\hat{Y}\right) - \sum_w \text{Var}\left(\hat{Y}_w\right) = \Lambda - C = \\ = \frac{M}{M-1}(C-D) - C = \\ = \frac{M}{M-1}\left(\frac{1}{M}C - D\right) = \\ = \frac{M}{M-1}\left(\frac{1}{M}\sum\sum_{w\neq v} Y_w Y_v - \sum\sum_{w\neq v}\sum_{i=1}^{M} y_{i,w}y_{v,i}\right) = \\ = -M\left(\sum_w\sum_v S_{w,v} - \sum_w S_w^2\right).\end{aligned} \tag{3.8}$$

The expression (3.8) is a correction term of the dependency of weekly samples. Interesting observation is that the correction term is a population parameter – it does not depend on a sample. From (3.7) and (3.8) follows that (3.4) for the current sampling design is equal to

$$\begin{aligned}\text{Var}\left(\hat{Y}\right) = M^2\frac{1-\frac{m}{M}}{m}\sum_w S_w^2 - M\left(\sum_w\sum_v S_{w,v} - \sum_w S_w^2\right) = \\ = \frac{M^2}{m}\sum_w S_w^2 - M\sum_w\sum_v S_{w,v}.\end{aligned} \tag{3.9}$$

It is interesting to observe that $\frac{M^2}{m}\sum_w S_w^2$ is the variance of $\hat{Y}$ if stratified simple random sampling with replacement stratified by weeks with the same sample size and allocation would be used.

3.5.2 Variance for Estimator of Ratio

The ratio of two population totals is

$$R = \frac{Y}{Z} = \frac{\sum_w Y_w}{\sum_w Z_w} = \frac{\sum_w \sum_{i=1}^{M} y_{i,w}}{\sum_w \sum_{i=1}^{M} z_{i,w}} = \frac{\sum_{k=1}^{N} y_k}{\sum_{k=1}^{N} z_k},$$

and the estimator of the ratio is

$$\hat{R} = \frac{\hat{Y}}{\hat{Z}} = \frac{\sum_w \hat{Y}_w}{\sum_w \hat{Z}_w} = \frac{\sum_w \sum_{i\in s} \frac{y_{i,w}}{\pi_{i,w}}}{\sum_w \sum_{i\in s} \frac{z_{i,w}}{\pi_{i,w}}} = \frac{\sum_{k\in s} \frac{y_k}{\pi_k}}{\sum_{k\in s} \frac{z_k}{\pi_k}}.$$

Approximate variance of $\hat{R}$ is derived. Taylor linearisation technique is applied to derive an approximate variance of $\hat{R}$ (Särndal et al., 1992, p. 178).

So called linearised variable $\boldsymbol{r}$ is introduced where the values of $\boldsymbol{r}$ are computed as

$$r_k = y_k - Rz_k,$$

or

$$r_{i,w} = y_{i,w} - Rz_{i,w}.$$

The structure of $\boldsymbol{r}$ is the same as for $\boldsymbol{y}$, $\boldsymbol{r} = (r_1, r_2, \ldots, r_k, \ldots, r_N)' = (\boldsymbol{r}'_1, \boldsymbol{r}'_2, \ldots, \boldsymbol{r}'_w, \ldots, \boldsymbol{r}'_W)'$, where $\boldsymbol{r}_w$ is a column vector of values $r_{i,w}$ at the week w. $\hat{R}$ is approximated as follows

$$\hat{R} \doteq \hat{R}_0 = R + \frac{1}{Z}\sum_{k=1}^{n}\frac{y_k - Rz_k}{\pi_k} = R + \frac{1}{Z}\sum_{k=1}^{n}\frac{r_k}{\pi_k},$$

where $\doteq$ stands for approximation. Approximate variance of $\hat{R}$ is given by

$$\text{AVar}\left(\hat{R}\right) = \text{Var}\left(\hat{R}_0\right) = \frac{1}{Z^2}\left(\sum_{k=1}^{N}\sum_{l=1}^{N}\frac{\pi_{kl}}{\pi_k\pi_l}r_k r_l - \left(\sum_{k=1}^{N} r_k\right)^2\right).$$

Approximate variance of $\hat{R}$ under the m-SSR design is expressed using (3.9)

$$\begin{aligned}\text{AVar}\left(\hat{R}\right) &= \frac{1}{Z^2}\left(\frac{M^2}{m}\sum_{w} S_w^2(r) - M\sum_{w}\sum_{v} S_{w,v}(r)\right) = \\ &= \left(\frac{M}{Z}\right)^2\left(\frac{1}{m}\sum_{w} S_w^2(r) - \frac{1}{M}\sum_{w}\sum_{v} S_{w,v}(r)\right),\end{aligned}$$

where $S_w^2(r)$ is the variance of $\boldsymbol{r}_w$ and $S_{w,v}(r)$ is the covariance of $\boldsymbol{r}_w$ and $\boldsymbol{r}_v$.

3.5.3 Variance under mSSRS

The population of units is split in non-overlapping strata in the case of stratified sampling. There are H strata, $V = \cup_{h=1}^{H} V_h$, and $M_h = |V_h|$ is the population size of stratum h, $\sum_{h=1}^{H} M_h = M$. The stratification of the unit population V determine the stratification of the element population U. The weekly sample size of stratum h is denoted by m_h, and $\sum_{h=1}^{H} m_h = m$. The vector $\boldsymbol{y}_h = (y_k : u_k \in U_h)$ is the vector of y_k values in stratum h, and $\boldsymbol{y}_{w,h} = (y_{i,w} : u_{i,w} \in U_{w,h})$ is the vector of $y_{i,w}$ values in stratum h. The variance of $\boldsymbol{y}_{w,h}$ is denoted by $S_{w,h}^2$ and the covariance of $\boldsymbol{y}_{w,h}$ and $\boldsymbol{y}_{v,h}$ is denoted by $S_{w,v,h}$.

The expression (3.9) is used to compute the variance of the estimate of total in stratum h,

$$\text{Var}\left(\hat{Y}_h\right) = \frac{M_h^2}{m_h}\sum_{w} S_{w,h}^2 - M_h\sum_{w}\sum_{v} S_{w,v,h}.$$

Variance of the estimate of the population total in the case of stratified sampling is equal to (Särndal et al., 1992, p. 102)

$$\text{Var}\left(\hat{Y}\right) = \sum_{h=1}^{H} \text{Var}\left(\hat{Y}_h\right) = \sum_{h=1}^{H} \left(\frac{M_h^2}{m_h} \sum_w S_{w,h}^2 - M_h \sum_w \sum_v S_{w,v,h} \right). \tag{3.10}$$

Approximate variance of the estimate of the ratio of two population totals in the case of stratified sampling is given by

$$\text{AVar}\left(\hat{R}\right) = \frac{1}{Z^2} \sum_{h=1}^{H} \left(\frac{M_h^2}{m_h} \sum_w S_{w,h}^2(r) - M_h \sum_w \sum_v S_{w,v,h}(r) \right),$$

where r is the so called linearised variable of R (more information regarding linearisation of ratio is available in Section 3.5.2).

Since $Y_q = \frac{1}{13} Y$, we have

$$\text{Var}\left(\hat{Y}_q\right) = \frac{1}{169} \sum_{h=1}^{H} \text{Var}\left(\hat{Y}_h\right) = \frac{1}{169} \sum_{h=1}^{H} \left(\frac{M_h^2}{m_h} \sum_w S_{w,h}^2 - M_h \sum_w \sum_v S_{w,v,h} \right),$$

and because of $R_q = R$, we have

$$\text{AVar}\left(\hat{R}_q\right) = \text{AVar}\left(\hat{R}\right).$$

3.6 Estimates by Monte Carlo Simulation

Section 3.6 is based on Mood, Graybill, and Boes (1974) unless otherwise stated.

Let X be a random variable with average μ and finite variance $\sigma^2 < \infty$. The sample $\boldsymbol{x} = (x_1, x_2, \ldots, x_m)$ is generated from X by Monte Carlo simulation. The sample $\boldsymbol{x}$ is treated as realisation of m independent and identically distributed (iid) random variables. The Monte Carlo estimate of μ is

$$\hat{\mu}_m = \bar{x}_m = \frac{1}{m} \sum_{i=1}^{m} x_i \tag{3.11}$$

The Monte Carlo estimate of σ^2 is

$$\hat{\sigma}_m^2 = s_m^2 = \frac{1}{m-1} \sum_{i=1}^{m} (x_i - \bar{x}_m)^2 \tag{3.12}$$

Both estimates (3.11) and (3.12) are unbiased estimates, $\text{E}(\bar{x}_m) = \mu$ and $\text{E}(s_m^2) = \sigma^2$.

3.6.1 Normally Distributed Variable

Let X be a normally distributed random variable with mean μ and variance σ^2. Then the distribution of $\bar{x}_m$ is normal $\bar{x}_m \sim N\left(\mu, \frac{\sigma^2}{m}\right)$. The variance of $\bar{x}_m$ is

$$\text{Var}\left(\bar{x}_m\right) = \frac{\sigma^2}{m}, \tag{3.13}$$

and the variance estimate is

$$\text{var}\left(\bar{x}_m\right) = \frac{s_m^2}{m}.$$

The distribution of s_m^2 is described by $\frac{(m-1)s_m^2}{\sigma^2} \sim \chi^2_{m-1}$. The variance of s_m^2 is

$$\text{Var}\left(s_m^2\right) = \frac{2\sigma^4}{m-1},$$

and the variance estimate is

$$\text{var}\left(s_m^2\right) = \frac{2s_m^4}{m-1}.$$

The random variables $\bar{x}_m$ and s_m^2 are independent by Cochran's theorem. The result $\frac{\bar{x}_m - \mu}{\frac{s_m}{\sqrt{m}}} \sim t_{m-1}$ follows from the distributions of $\bar{x}_m$ and s_m^2 and the independence of $\bar{x}_m$ and s_m^2.

A two-sided 100α percent confidence interval for μ is constructed as

$$p\left(\bar{x}_m - \frac{s_m}{\sqrt{m}} t_{m-1,1-\frac{\alpha}{2}} < \mu < \bar{x}_m + \frac{s_m}{\sqrt{m}} t_{m-1,1-\frac{\alpha}{2}}\right) = 1 - \alpha.$$

A one-sided 100α percent confidence interval for σ^2 is constructed as

$$p\left(\sigma^2 < \frac{(m-1)s_m^2}{\chi^2_{m-1,\alpha}}\right) = 1 - \alpha. \tag{3.14}$$

3.6.2 Unspecified Distribution of Variable

Let X be a random variable with mean μ and finite variance σ^2 but the distribution of X is not specified. The distribution of $\bar{x}_m$ tends to the normal distribution as m tends to infinity $\bar{x}_m \xrightarrow[m\to\infty]{} N\left(\mu, \frac{\sigma^2}{m}\right)$ by the Central limit theorem.

The variance of $\bar{x}_m$ is expressed by (3.13). It is because the elements of $\boldsymbol{x}$ are iid random variables. The variance of s_m^2 is

$$\text{Var}\left(s_m^2\right) = \frac{1}{m}\left(\mu_4 - \frac{m-3}{m-1}\sigma^4\right).$$

where μ_4 is the fourth central moment of the X (Mood et al., 1974, p. 229).

3.6.3 Nonparametric Bootstrap

Section 3.6.3 is based on Wasserman (2006) unless otherwise stated.

Nonparametric bootstrap can be used to construct bootstrap percentile intervals for the estimates of Monte Carlo simulations. Bootstrap sample is a sample selected with replacement from $\boldsymbol{x}$ with sample size m. Several bootstrap samples are drawn, the number of bootstrap samples drawn is denoted by J. Bootstrap samples are labelled with the index $j = \{1, 2, \ldots, J\}$. j-th bootstrap sample is denoted by $\tilde{\boldsymbol{x}}_j$. The elements of the j-th bootstrap sample are denoted by $\tilde{\boldsymbol{x}}_j = (\tilde{x}_{j1}, \tilde{x}_{j2}, \ldots, \tilde{x}_{jm})$, $\tilde{x}_{ji} \in \boldsymbol{x}$ for all j and for all i.

The estimate of μ computed from the j-th bootstrap sample is

$$\bar{x}_{mj} = \frac{1}{m} \sum_{i=1}^{m} \tilde{x}_{ji}.$$

The estimate of σ^2 computed from the j-th bootstrap sample is

$$s^2_{mj} = \frac{1}{m-1} \sum_{i=1}^{m} \left(\tilde{x}_{ji} - \hat{\mu}_j\right)^2.$$

The distribution of the vector $\bar{\boldsymbol{x}}_{mb} = (\bar{x}_{m1}, \bar{x}_{m2}, \ldots, \bar{x}_{mJ})$ of J bootstrap estimates of μ asymptomatically approximates the distribution of $\bar{x}_m$. The distribution of the vector $\boldsymbol{s}^2_{mb} = (s^2_{m1}, s^2_{m2}, \ldots, s^2_{mJ})$ asymptomatically approximates the distribution of s^2_m.

A two-sided bootstrap percentile interval for μ is constructed as

$$p\left(\hat{Q}_{\frac{\alpha}{2}}\left(\bar{\boldsymbol{x}}_{mb}\right) < \mu < \hat{Q}_{1-\frac{\alpha}{2}}\left(\bar{\boldsymbol{x}}_{mb}\right)\right) \approx 1 - \alpha,$$

where $\hat{Q}_{\frac{\alpha}{2}}\left(\bar{\boldsymbol{x}}_{mb}\right)$ and $\hat{Q}_{1-\frac{\alpha}{2}}\left(\bar{\boldsymbol{x}}_{mb}\right)$ are the estimates of quantiles at probabilities $\frac{\alpha}{2}$ and $1-\frac{\alpha}{2}$ of the unknown distribution of $\bar{x}_m$. The quantiles are estimated from the vector of bootstrap estimates $\bar{\boldsymbol{x}}_{mb}$. A one-sided bootstrap percentile interval for σ^2 is constructed similarly

$$p\left(\sigma^2 < \hat{Q}_{1-\alpha}\left(\boldsymbol{s}^2_{mb}\right)\right) \approx 1 - \alpha, \tag{3.15}$$

where $\hat{Q}_{\alpha}\left(\boldsymbol{s}^2_{mb}\right)$ is the estimate of α-quantile of the unknown distribution of s^2_m. The quantile is estimated from the vector of bootstrap estimates $\boldsymbol{s}^2_{mb}$.

Confidence bands are used to monitor the variance of the Monte Carlo simulation results. A 100α percent confidence band is constructed by computing a 100α percent confidence interval for each simulation iteration k (Robert & Casella, 2004, 2010).

Chapter 4

Cost Efficiency – Practical Application

The cost efficiency of three sampling designs is analysed in the chapter. Fieldwork cost modelling, travelling salesman problem solving, Monte Carlo simulation techniques, survey sampling techniques and hypothesis testing are used to provide the results of the chapter.

4.1 Fieldwork Cost Estimation

The research of survey field operations is a brand new topic in the scope of statistical research. Chen (2008) is writing about the research of survey field operations:

> So far, similar work is rarely found in the literature describing the analytical or simulation modeling of the operations. The field operation is a unique system in the operations research field.

The field operations of surveys can be classified as stochastic dynamic systems. Usually the field operations cannot be modelled analytically because of the complexity of the system (Cox, 2012). Discrete-event simulation modelling could be a tool for performance evaluation of survey field research.

The aim of this section is to estimate the fieldwork cost of the survey. Sampled dwellings assigned to interviewer are split by weeks. Cost is expressed in Latvian currency – Latvian lats. Two components of fieldwork cost are assumed:

- travel cost,
- interview cost.

Travel cost is the most complicated component to be estimated. Estimation of travel cost is done in several steps:

- simplified model for travel cost is developed,
- expected travel cost for two-stage sampling design computed by the simplified model is estimated using simulation,
- adjustment of the simplified model is made if necessary to achieve modelled cost to be approximately close to real cost.

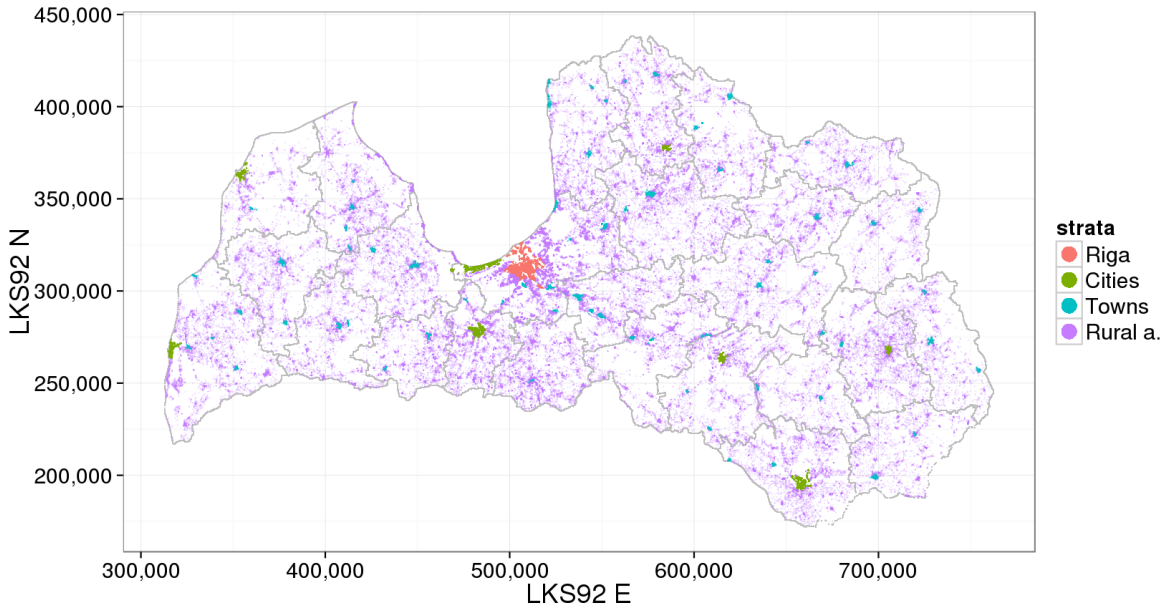


Figure 4.1 **The scatter plot of the geographical coordinates of dwellings in the population frame**

Several assumptions are made to develop a simplified model for interview cost:

- data collection is done by personal interviews,
- full response model is assumed,
- full response is achieved at the first visit of the dwelling – revisiting of dwellings is not done,
- interviewer visits all assigned dwellings for a week once,
- trip done by interviewer starts at the residence of interviewer, goes through all assigned dwellings by shortest distance and ends at the residence of interviewer,
- interviewing for a weekly sample can be done in the time of a week – a trip never takes more time than a week,
- car is a mode of transport for interviewers for all distances.

The function

$$c_1(s) = dK_f C_f K_d \tag{4.1}$$

is provided as a simplified model for modelling travel cost where d is the total travelling distance done by all interviewers expressed in kilometres, K_f is average fuel consumption expressed in l/km, C_f is an average price of fuel expressed in Ls/l and K_d is an adjustment coefficient specified by a user (default value of K_d is 1).

The information available for modelling travel distance is the geographical coordinates of all dwellings in the population frame and the geographical coordinates of all residences of interviewers. Figure 4.1 shows all dwellings mapped on the scatter plot.

The total travel distance is computed as

$$d = \sum_{g=1}^{G} \sum_{w=1}^{W} d_{g,w},$$

where G is the number of interviewers, W is the number of weeks observed, $d_{g,w}$ is the length of the path done by the interviewer g during the field work operation in the week w. The $d_{g,w}$ is computed by the help of the travelling salesperson problem (TSP) (Hahsler & Hornik, 2007). The input arguments for the procedure to compute $d_{g,w}$ are:

- matrix $\boldsymbol{A}_{g,w}$ with two columns where each row of the matrix represents the coordinates of sampled dwellings assigned to the interviewer g from the sample of week w, the number of rows is denoted by $n_{g,w}$,
- vector $\boldsymbol{v}_g$ (length two) with the coordinates of the residence of interviewer g.

Matrix $\boldsymbol{B}_{g,w}$ is constructed by attaching $\boldsymbol{v}'_g$ to $\boldsymbol{A}_{g,w}$ as extra row

$$\boldsymbol{B}_{g,w} = \begin{pmatrix} \boldsymbol{v}'_g \\ \boldsymbol{A}_{g,w} \end{pmatrix}.$$

The size of $\boldsymbol{B}_{g,w}$ is 2 columns and $n_{g,w} + 1$ rows.

Rows of the $\boldsymbol{B}_{g,w}$ define the set of nodes for TSP. Distances between nodes are computed as euclidean distances (Chen, 2008, p. 16). TSP is defined by the nodes and the distance function. The TSP is solved by the nearest insertion algorithm (Rosenkrantz, Stearns, & Lewis, 1977, p. 572). The result of the nearest insertion algorithm is not necessarily optimal. The tour length obtained by the nearest insertion algorithm is shorter than doubled length of the optimal tour. It is proved by Rosenkrantz et al. (1977, p. 573). Practical solving of a TSP is implemented by the following steps.

1. The $\boldsymbol{D}_{g,w}$ is a square matrix with size $(n_{g,w} + 1) \times (n_{g,w} + 1)$, where $d_{i,j}$ is an euclidean distance between the points represented by the rows i and j of the $\boldsymbol{B}_{g,w}$. All diagonal elements of $\boldsymbol{D}_{g,w}$ are equal to 0, $\forall d_{i,i} = 0$. The lower triangle of $\boldsymbol{D}_{g,w}$ is computed from $\boldsymbol{B}_{g,w}$ by the R function `dist` (R Core Team, 2013).
2. Symmetric TSP (Hahsler & Hornik, 2007, p. 2) is created by the R function `TSP` from the package `TSP` (Hahsler & Hornik, 2011) using the lower triangle of the $\boldsymbol{D}_{g,w}$ as an argument.
3. The TSP is solved by the R function `solve_TSP` from the package `TSP` (Hahsler & Hornik, 2011) using the nearest insertion algorithm. The residence of interviewer is set as the first node of the tour.
4. The attribute `tour_length` is extracted from the result of the function `solve_TSP`. The value of the attribute `tour_length` is taken as a value of $d_{g,w}$. The $d_{g,w}$ is expressed in metres.

Interview cost is computed by the function $c_2(s) = mC_h + nC_p$ where m is number of dwellings in the sample s, n is the number of individuals in the sample s, C_h is the interview cost for a household questionnaire, C_p is the interview cost for a individual questionnaire.

The cost function

$$c(s) = c_1(s) + c_2(s) = K_f C_f K_d \sum_{g=1}^{G} d_g + mC_h + nC_p \tag{4.2}$$

is used to measure the cost of the survey in Monte Carlo simulations.

4.2 Procedures for Monte Carlo Simulations

There is a number of R (R Core Team, 2013) procedures (functions by R conception) developed by the author to run Monte Carlo simulations. The description and code of procedures is given in Appendix 1. The code of procedures and additional code related to running the Monte Carlo simulations are available online at "GitHub" repository (Liberts, 2013b). Please refer to the online repository for the most up-to-date version of the code.

1. Sample generation functions:
   (a) Simple random sampling
   (b) Simple random sampling with even distribution of sampled units by weeks
   (c) Stratified Simple random sampling with even distribution of sampled units by weeks
   (d) Cluster sampling
   (e) Stratified cluster sampling
   (f) Cluster sampling with even distribution of sampled units by weeks
   (g) Stratified Cluster sampling with even distribution of sampled units by weeks
   (h) Two-stage sampling
2. The calculation of interviewing expenses:
   (a) TSP solver for single interviewer
   (b) TSP solver for multiple interviewers
   (c) The calculation of interviewing expenses
3. The estimation of population parameters:
   (a) The estimation of the primary population parameters from sample data
   (b) The estimation of the secondary population parameters
4. Other functions:
   (a) The extraction of data from the dynamic population data according to the sampled individuals and weeks
   (b) Monte Carlo simulations

4.3 Cost of Two-Stage Sampling Design

The survey budget γ has to be set to evaluate the cost efficiency by Definition 2. The field work budget γ is set equal to the survey budget necessary to run the LFS with current sampling design (see Chapter 1) for a quarter. The aim of the first phase simulation is to estimate the expected field work cost for the LFS with the two-stage sampling design. The expected total field work cost and the expected field work cost allocation by three strata ("Riga", "Cities" and "Towns and rural areas") are estimated.

4.3.1 Information Available from the Real LFS

Some information about the LFS cost in 2010 is available. The information available is based on some assumptions. The information available does not necessarily conform with the real situation, though it can be used for this research.

The travel distance done by interviewers for LFS in 2010 has been 191 063 km. It makes travel distance equal to 47 766 km per quarter on average. The travel cost for LFS in 2010 has been 13 874 Ls. It makes travel cost equal to 3468 Ls per quarter on average. The average retail price of Gasoline A-95 and Diesel fuel in 2010 has been 0.768 Ls/l and 0.752 Ls/l (Central Statistical Bureau of Latvia, 2012b). It makes the average price of fuel equal to 0.760 Ls/l under the assumption of equal share of Gasoline A-95 and Diesel fuel used by interviewers. The amount of fuel consumed by interviewers on average per quarter was 4564 l in 2010. It makes the average fuel consumption equal to 0.0955 l/km.

Interviewers get paid approximately three lats 3 Ls for a completed household questionnaire and 1 Ls for a completed individual questionnaire.

4.3.2 Estimation of Field Work Budget

The first simulation experiment is done to estimate the expected fieldwork cost for the current sampling design used for LFS. The results are compared with the information available from the real survey. The adjustment of the cost function is introduced.

Simulation setting:

- Design: Two-stage sampling design
- Design parameters:

s	A	B	W	d	Q	w	M	m
1	8	1	13	0	6	13	256,556	10
2	8	2	13	0	2	13	157,709	7
3	8	2	13	0	7	13	129,823	8
4	8	2	13	0	11	13	198,515	9

- The number of iterations: 6000

- Estimates to be calculated each iteration:

Parameter	Description
n_{0k}	the total number of individuals in sample
n_{1k}, n_{2k}, n_{3k}	the number of individuals in sample by strata
m_{0k}	the total number of households in sample
m_{1k}, m_{2k}, m_{3k}	the number of households in sample by strata
d_{0k}	the total distance done by interviewers
d_{1k}, d_{2k}, d_{3k}	the distance done by interviewers by strata

There are four strata defined by sampling design. The stratum "towns" ($h = 3$) and the stratum "rural areas" ($h = 4$) is merged during the cost estimation. The merged strata are defined by $h = 3$. It is done because the same interviewers are working in both strata. It is possible that an interviewer is doing a field work in both strata during a week. Therefore the travelling cost can not be separated by both strata. There are separate interviewers in the strata "Riga" ($h = 1$) and "Cities" ($h = 2$).

n_{0k}, m_{0k} and d_{0k} are computed for a verification purpose. The following equalities are verified.

$$\sum_{h=1}^{3} n_{hk} = n_{0k}$$

$$\sum_{h=1}^{3} m_{hk} = m_{0k}$$

$$\sum_{h=1}^{3} d_{hk} = d_{0k}$$

Simulation of Distance

The main characteristics of the simulated travel distances are displayed in Table 4.1. All results are displayed for domains: "Riga", "Cities", "Towns & Rural" – towns and rural areas, and "Latvia" – the whole population. There are mean distance $\bar{x}_n$, standard deviation of distance s_n, the p-value of the Anderson-Darling test (AD) for normality (Anderson & Darling, 1952) and the p-value of the Lilliefors test (Li) for normality (Lilliefors, 1967). Density plots and Quantile-Quantile plots (Q-Q plots) of the simulated distances by domains are in the first and second columns of Figure 4.2.

The null hypothesis (the distribution is normal) is rejected at the 5% level of significance (all p-values of the tests are less than 5%). The density and Q-Q plots clearly show the density of distance in Riga and Cities is not normally distributed.

The 99% confidence bands based on normal distribution and 99% bootstrap confidence bands are plotted in the third column of Figure 4.2. 2000 bootstrap replicates were used. Note the 99% confidence bands (based on normal distribution) are centred on $\bar{x}_n$, not on $\bar{x}_k$. It is done

Table 4.1

The summary statistics of simulated travel distance

domain	$\bar{x}_n$	s_n	AD p-value	Li p-value
Riga	296.578	11.072	0.000	0.000
Cities	912.865	20.684	0.000	0.000
Towns and rural areas	12981.201	142.833	0.000	0.001
Latvia	14190.644	145.324	0.000	0.001

Table 4.2

The 99% confidence intervals for expected travel distance

domain	$\bar{x}_n$	CI normal		CI boot	
Riga	296.6	296.2	296.9	296.2	297.0
Cities	912.9	912.2	913.6	912.1	913.6
Towns and rural areas	12981.2	12976.4	12986.0	12976.7	12986.0
Latvia	14190.6	14185.8	14195.5	14185.6	14195.6

for easier comparison of both bands. It is possible to observe that both bands are quite close to each other.

The precision estimates for the estimates of expected travel distance expressed as confidence intervals are given in Table 4.2. There are confidence interval based on normal distribution (CI normal) and bootstrap confidence interval (CI boot).

The mean simulated distance $d_0 = \bar{d}_{0k} = \frac{1}{n}\sum_{k=1}^{n} d_{0k}$ is 14 191 km. The observed distance in the real LFS 2010 $\tilde{d}_0$ was 47 766 km. The difference is significant. The main reasons of the difference is the assumptions used for the simulation (defined in Section 4.1). The adjustment coefficient K_d for the cost function (4.2) is computed as $K_d = \frac{\tilde{d}_0}{d_0}$.

Travel cost is computed by (4.1). The precision estimates for expected travel cost are presented in Table 4.3.

Simulation of Interview Cost

The results of the simulated interview cost are in Table 4.4. The normality of interview cost can not be rejected (at the 5% level of significance) by the Anderson-Darling test. The density

Table 4.3

The 99% confidence intervals for expected travel cost

domain	$\bar{x}_n$	CI normal		CI boot	
Riga	90.6	90.5	90.7	90.5	90.7
Cities	278.8	278.6	279.0	278.5	279.0
Towns and rural areas	3964.2	3962.8	3965.7	3962.7	3965.7
Latvia	4333.6	4332.1	4335.0	4332.0	4335.0

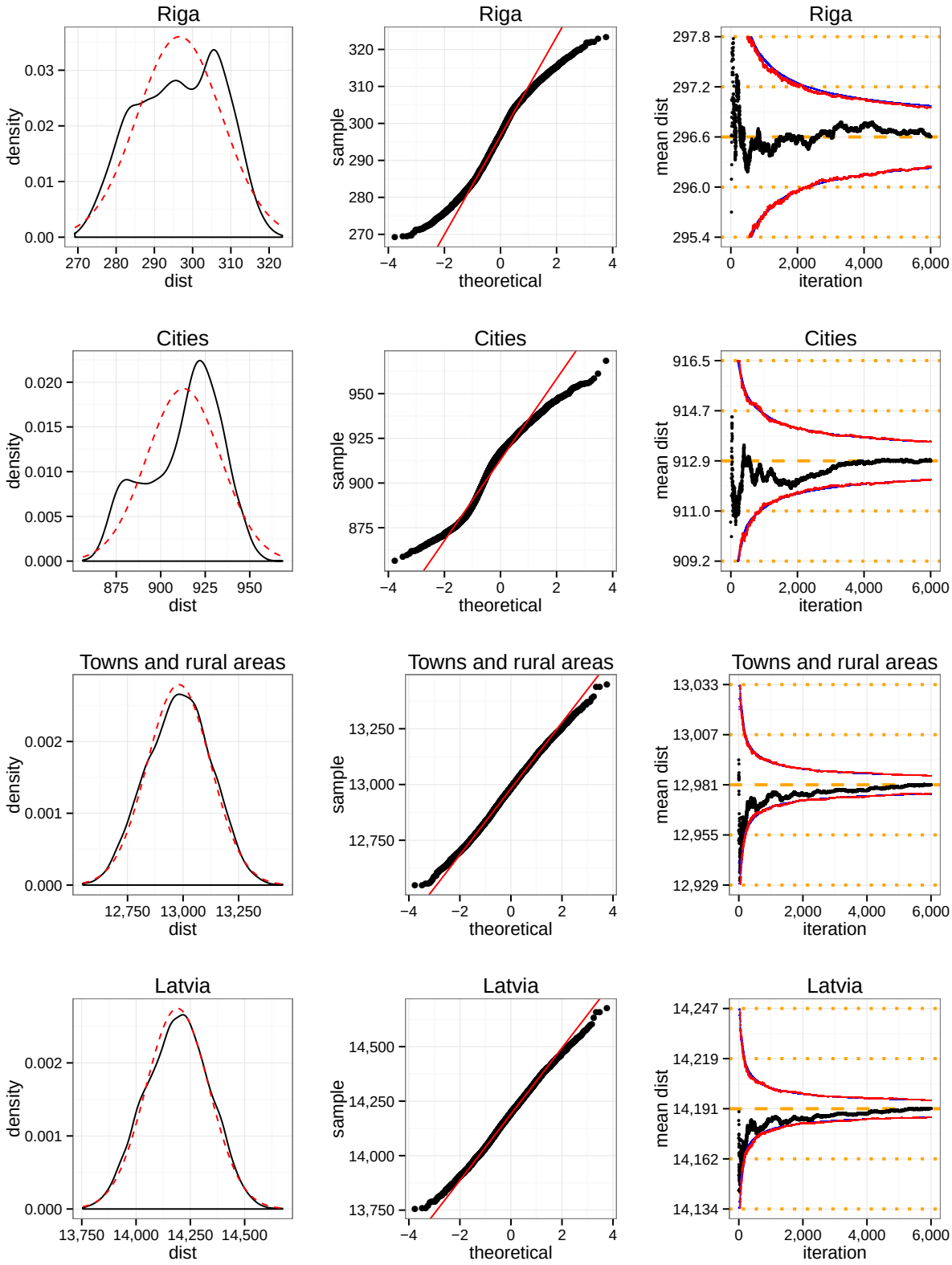


Figure 4.2 **Density plots, Q-Q plots, MC convergence plots; variable: distance**

Table 4.4

The summary statistics of simulated interview cost

domain	$\bar{x}_n$	s_n	AD p-value	Li p-value
Riga	5304.572	38.900	0.222	0.014
Cities	7440.758	44.598	0.126	0.036
Towns and rural areas	18925.870	87.352	0.624	0.150
Latvia	31671.199	105.391	0.808	0.589

Table 4.5

The 99% confidence intervals for expected interview cost

domain	$\bar{x}_n$	CI normal		CI boot	
Riga	5304.6	5303.3	5305.9	5303.3	5305.9
Cities	7440.8	7439.3	7442.2	7439.3	7442.3
Towns and rural areas	18925.9	18923.0	18928.8	18922.9	18928.7
Latvia	31671.2	31667.7	31674.7	31667.4	31674.8

plots, Q-Q plots and MC convergence plots of simulated interview cost are in Figure 4.3. The estimates of the 99% confidence intervals for expected interview cost are in Table 4.5.

Fieldwork Cost

The simulated travel cost and interview cost are added to compute the estimates of total fieldwork cost. See the details in Table 4.6, Table 4.7 and Figure 4.4. The normality of fieldwork cost can not be rejected (at the 5% level of significance) by both tests.

The estimated value of the expected fieldwork cost for the current sampling design used for LFS is 36 004.8 Ls with the 99% confidence interval $(36\,001.0; 36\,008.5)$. The total survey budget γ is taken to be equal to 36 004.8 Ls. The expected allocation of the fieldwork cost by three strata (γ_1, γ_2 and γ_3) is taken according to Table 4.7.

Table 4.6

The summary statistics of simulated fieldwork cost

domain	$\bar{x}_n$	s_n	AD p-value	Li p-value
Riga	5395.141	39.858	0.586	0.695
Cities	7719.531	45.809	0.489	0.348
Towns and rural areas	22890.098	95.524	0.323	0.308
Latvia	36004.770	113.175	0.511	0.626

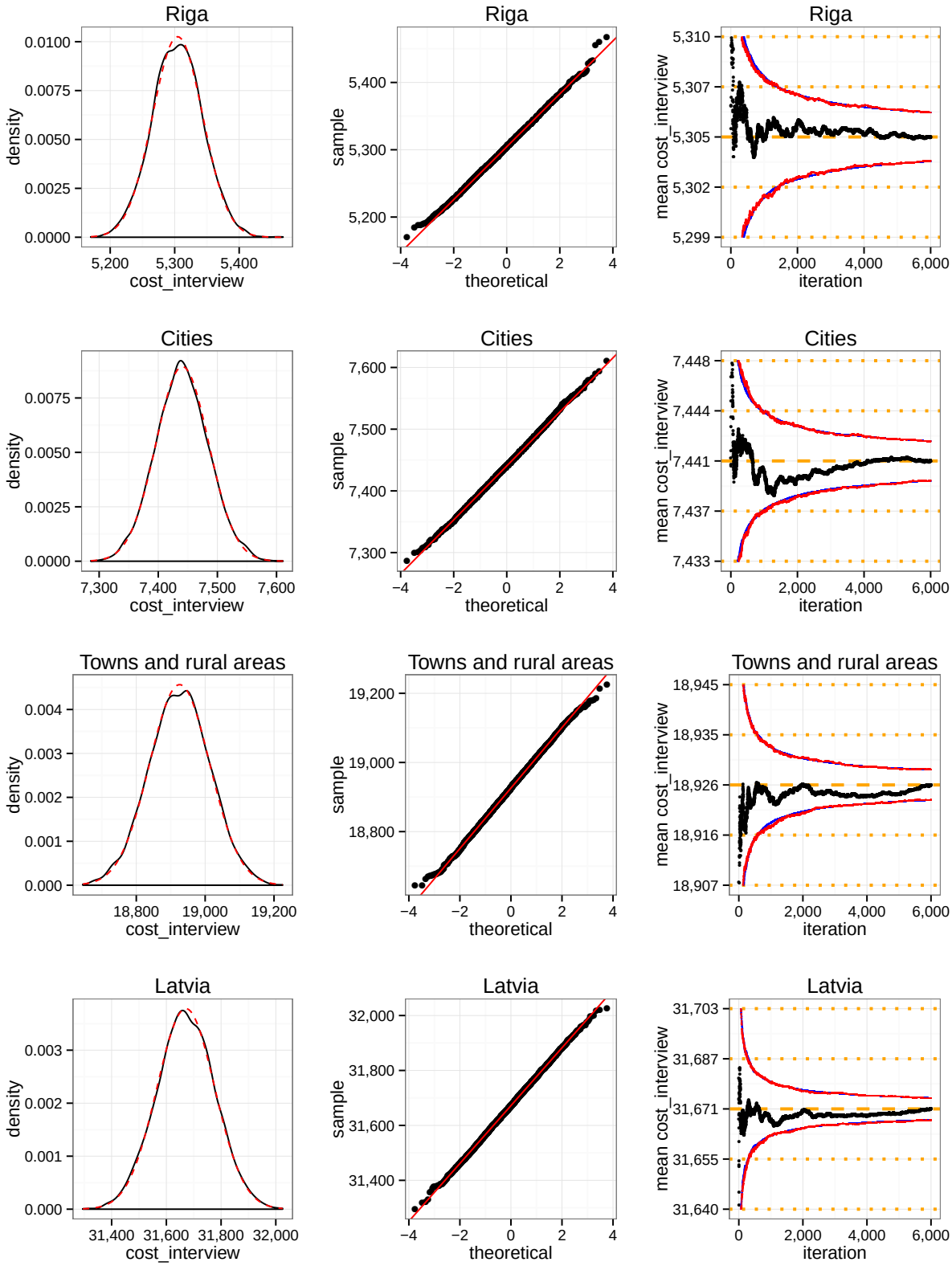


Figure 4.3 **Density plots, Q-Q plots, MC convergence plots; variable: interview cost**

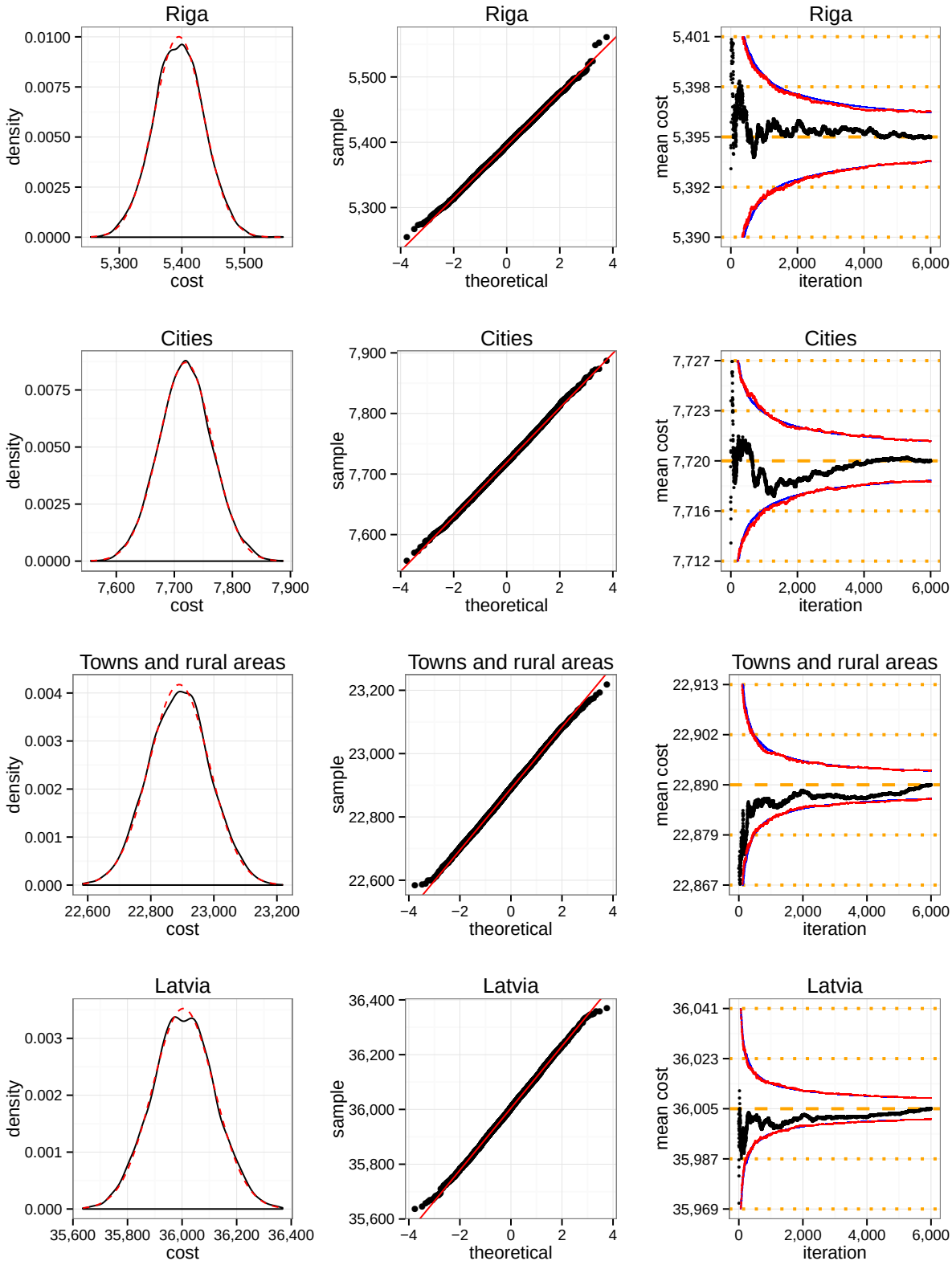


Figure 4.4 **Density plots, Q-Q plots, MC convergence plots; variable: fieldwork cost**

Table 4.7

The 99% confidence intervals for expected fieldwork cost

domain	$\bar{x}_n$	CI normal		CI boot	
Riga	5395.1	5393.8	5396.5	5393.9	5396.5
Cities	7719.5	7718.0	7721.1	7717.9	7721.0
Towns and rural areas	22890.1	22886.9	22893.3	22886.9	22893.3
Latvia	36004.8	36001.0	36008.5	36001.1	36008.6

4.4 Sample Size of the Alternative Designs

The aim of the second phase simulation is to estimate sample size and sample allocation by three strata for the two other designs (see Section 3.3) so that expected fieldwork cost allocated by three strata for these designs would be approximately equal to γ_1, γ_2 and γ_3 (Table 4.7).

4.4.1 Expected Cost and Sample Size

There are two designs and three strata for each design. The sample size is estimated independently for each design and stratum (six cases). Modified simple random sampling of individuals or households is done in each stratum. Stratum sample size n_h is the only variable parameter here, $s_h = s_h(n_h)$. The valid values of n_h are $n_h : (0 < n_h \leq N_h \;\&\; n_h \bmod 13 = 0)$ where N_h is the population size of the stratum h.

The relation between the expected cost and sample size has to be studied to find the necessary sample sizes. The cost function is defined by (4.2)

$$c(s_h) = K_f C_f K_d d(s_h) + C_h m_h(s_h) + C_p m_p(s_h)$$

where K_f, C_f, K_d, C_h and C_p are constants. d (distance), m_h (the number of household questionnaires) and m_p (the number of individual questionnaires) are random variables depending on sample s_h. $d = d(s_h)$, $m_h = m_h(s_h)$ and $m_p = m_p(s_h)$. Expected values of d, m_h and m_p depend on s_h. $\mathrm{E}(d) = D = D(s_h)$, $\mathrm{E}(m_h) = M_h = M_h(s_h)$ and $\mathrm{E}(m_p) = M_p = M_p(s_h)$. The expected cost is expressed as

$$\mathrm{E}(c(s_h)) = C(s_h) = K_f C_f K_d D(s_h) + C_h M_h(s_h) + C_p M_p(s_h).$$

The expected cost can be rewritten as the function of n_h because $s_h = s_h(n_h)$

$$\mathrm{E}(c(n_h)) = C(n_h) = K_f C_f K_d D(n_h) + C_h M_h(n_h) + C_p M_p(n_h).$$

All three functions $D(n_h)$, $M_h(n_h)$ and $M_p(n_h)$ are monotonically increasing functions. The statement that $C(n_h)$ is a monotonically increasing function follows from the monotony of the three functions.

The aim is to find n_h so that $C(n_h) \approx \gamma_h$ where γ_h is the budget for stratum h. The solution is defined as

$$n_h^* = \underset{\{n_h : C(n_h) > \gamma_h\}}{\arg\min} \; C(n_h).$$

The solution belongs to the set $\boldsymbol{n}_h = \left\{13, 26, 39, \ldots, n_h^{TSSh}\right\}$ where n_h^{TSSh} is the expected sample size (individuals or households) under two-stage sampling.

4.4.2 Sample Size Estimation

The first step is to approximate the relation between n_h and $C(n_h)$. Eight evenly distributed sample sizes are selected from the set $\boldsymbol{n}_h$ and expected cost is estimated with each selected sample size. The estimation of the expected cost is done by Monte Carlo simulation for each sampling design and each stratum. The eight selected points are plotted in Figure 4.5.

The relation can be described with non-linear regression in the form of

$$C(n_h) \sim \beta_0 + \beta_1 n_h + \beta_2 \sqrt{n_h}$$

where $C(n_h)$ is the expected cost and n_h is sample size. The estimates of the regression coefficients are computed by the function `lmList` from the R package `nlme` (Pinheiro, Bates, DebRoy, Sarkar, & R Core Team, 2012). The fitted regression line is added to the plots in Figure 4.5. The green horizontal line represents the budget line γ_h.

The possible value of n_h^* is computed from the regression equation

$$\gamma_h = \hat{\beta}_0 + \hat{\beta}_1 \hat{n}_h^* + \hat{\beta}_2 \sqrt{\hat{n}_h^*}. \tag{4.3}$$

The solution of (4.3) is

$$\hat{n}_h^* = \frac{\left(\sqrt{\hat{\beta}_2{}^2 - 4\hat{\beta}_1\left(\hat{\beta}_0 - \gamma_h\right)} - \hat{\beta}_2\right)^2}{4\hat{\beta}_1{}^2}.$$

The point $(\hat{n}_h^*, \gamma_h)$ is marked in the plots with blue cross.

The second step is to estimate the cost for sample sizes around $\hat{n}_h^*$ to find the solution n_h^*. Seven sample sizes are selected for the simulation: $\hat{n}_h^* - 39, \hat{n}_h^* - 26, \hat{n}_h^* - 13, \hat{n}_h^*, \hat{n}_h^* + 13, \hat{n}_h^* + 26, \hat{n}_h^* + 39$.

It is obvious from the plots in Figure 4.6 that selected seven sample sizes contains the solution. The solution is marked with green circle. The confidence intervals are drawn for expected cost (in red colour). The numeric results are shown in Table 4.8. Stratified random sampling design of individuals is denoted by `SRS` and stratified random sampling design of households is denoted by `Cluster` in the table.

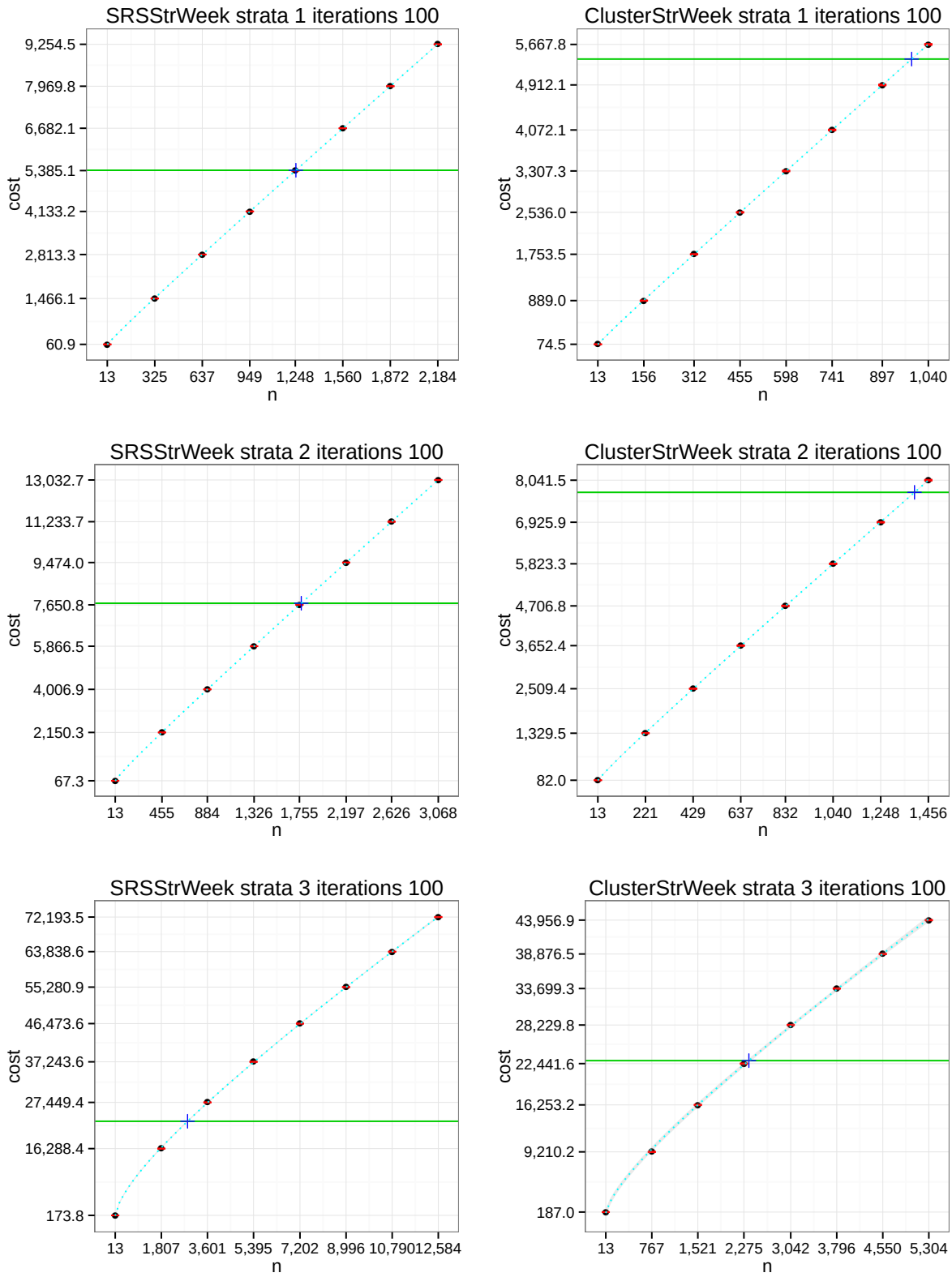


Figure 4.5 **The relation of expected cost and sample size**

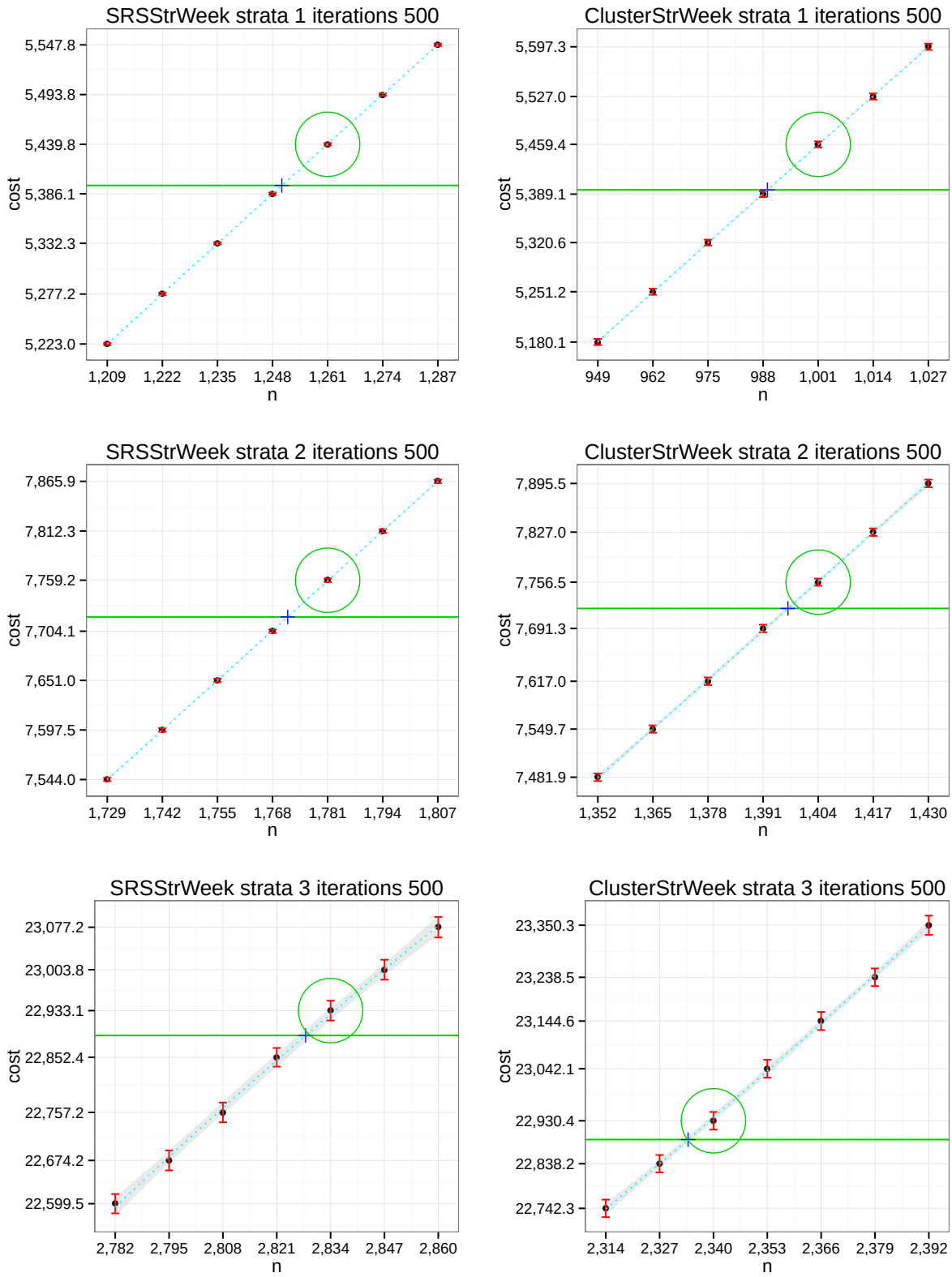


Figure 4.6 **Sample size in line with defined fieldwork budget**

Table 4.8

The sample allocation in line with defined fieldwork budget

design	strata	n	cost	cost.sd	cost.cil	cost.ciu	γ_h
SRSStrWeek	1	1261	5439.8	11.6	5438.5	5441.1	5395.1
SRSStrWeek	2	1781	7759.2	15.9	7757.3	7761.0	7719.5
SRSStrWeek	3	2834	22933.1	148.0	22916.0	22950.2	22890.1
ClusterStrWeek	1	1001	5459.4	36.9	5455.1	5463.6	5395.1
ClusterStrWeek	2	1404	7756.5	43.3	7751.5	7761.6	7719.5
ClusterStrWeek	3	2340	22930.4	163.4	22911.5	22949.3	22890.1

4.5 Precision of Population Parameter Estimates

The aim of the section is to compute the precision of estimates under the considered sampling designs and under condition of the approximately equal expected cost. The precision of the estimates is expressed as their variance. The variance is necessary for the estimates of two population parameters (3.1) and (3.2).

Variance of the estimates under mSSRSi and mSSRSh is calculated using (3.10). The numerical computation of variance was supported by the R package `data.table` (Dowle, Short, & Lianoglou, 2012) because of the high dimension of the population data (the dimension of a matrix for individual economic status over 13 weeks is 1 651 126 $\times$ 13).

Monte Carlo simulation experiment is used to estimate the variance of the estimates under the two-stage sampling design. Monte Carlo simulation experiment is chosen because it is more efficient technique in this case compared to analytical derivation of variance for estimates under this design (the main complication is computation of $\boldsymbol{\Pi}$ for the current design).

The values of $\hat{Y}_q$ and $\hat{R}_q$ are computed each iteration. $\widehat{\text{Var}}\left(\hat{Y}_q\right)$ and $\widehat{\text{Var}}\left(\hat{R}_q\right)$ are computed by (3.12). Confidence interval for $\text{Var}\left(\hat{Y}_q\right)$ and $\text{Var}\left(\hat{R}_q\right)$ are constructed by (3.14) and (3.15).

The simulation is done with the same sampling design parameters as in the simulation for estimating the cost of two-stage sampling design (Section 4.3). The number of iterations is 20 000.

4.6 Results of Cost Efficiency Analysis

90 population parameters (45 totals and 45 ratios of two totals) were selected for the cost efficiency analysis (see Table 4.9). There are parameters representing the whole population and also population domains. Two sets of domains are considered:

- strata (4) – Riga, cities, towns and rural areas,
- age group (2) – individuals aged 15–24 and 25–74 years.

The three selected designs are compared by cost efficiency using Definition 2 for estimation of each population parameter. Hypothesis testing is used for comparing variance under TSSh

Table 4.9

Population parameters chosen for the cost efficiency analysis

Parameter	Notation	Domain	Count
Number of employed individuals	t.empl	All	1
Number of unemployed individuals	t.unempl	All	1
Number of inactive individuals	t.inact	All	1
Number of employed individuals	t.empl	Strata (4)	4
Number of unemployed individuals	t.unempl	Strata (4)	4
Number of inactive individuals	t.inact	Strata (4)	4
Number of employed individuals	t.empl	Age group (2)	2
Number of unemployed individuals	t.unempl	Age group (2)	2
Number of inactive individuals	t.inact	Age group (2)	2
Number of employed individuals	t.empl	Strata (4) × age group (2)	8
Number of unemployed individuals	t.unempl	Strata (4) × age group (2)	8
Number of inactive individuals	t.inact	Strata (4) × age group (2)	8
Activity rate	r.act	All	1
Employment rate	r.empl	All	1
Unemployment rate	r.unempl	All	1
Activity rate	r.act	Strata (4)	4
Employment rate	r.empl	Strata (4)	4
Unemployment rate	r.unempl	Strata (4)	4
Activity rate	r.act	Age group (2)	2
Employment rate	r.empl	Age group (2)	2
Unemployment rate	r.unempl	Age group (2)	2
Activity rate	r.act	Strata (4) × age group (2)	8
Employment rate	r.empl	Strata (4) × age group (2)	8
Unemployment rate	r.unempl	Strata (4) × age group (2)	8

design with variance under mSSRSi or mSSRSh. Assumption is made that the estimates of population parameters under TSSh design are normally distributed:

$$\hat{\theta} \sim N\left(\mu, \sigma^2\right),$$

where σ^2 is unknown. It is estimated by $s^2 = s^2(\boldsymbol{x})$ from data of simulation experiment denoted by $\boldsymbol{x}$. The length of $\boldsymbol{x}$ is equal to the number of iteration in simulation, $|\boldsymbol{x}| = n = 20000$ in this case. The aim is to compare σ^2 by TSSh design with known σ_0^2 by another design. One-sided hypothesis test according to Wasserman (2004) is done:

$$\begin{aligned} H_0 &: \sigma^2 \geq \sigma_0^2, \\ H_1 &: \sigma^2 < \sigma_0^2. \end{aligned} \tag{4.4}$$

Test statistic is computed as

$$T(\boldsymbol{x}) = \frac{(n-1)\, s^2}{\sigma_0^2}.$$

Rejection region R is defined as

$$R = \{\boldsymbol{x} : T(\boldsymbol{x}) \leq c\},$$

where $c = F_{n-1}^{-1}(\alpha)$ is the value of the inverse cumulative distribution function of χ_{n-1}^2 distribution at α. The following statements regarding H_0 are set:

$$\begin{aligned} &T(\boldsymbol{x}) \leq c \Rightarrow \text{reject } H_0, \\ &T(\boldsymbol{x}) > c \Rightarrow \text{retain (do not reject) } H_0. \end{aligned}$$

The type I error to reject H_0 when H_0 is true is less or equal to α.

$$\begin{aligned} p(T(\boldsymbol{x}) \leq c \mid H_0) &= p\left(\frac{(n-1)\, s^2}{\sigma_0^2} \leq c \,\middle|\, H_0\right) = \\ &= p\left(\frac{(n-1)\, s^2}{\sigma^2} \leq \frac{\sigma_0^2}{\sigma^2} c \,\middle|\, H_0\right) \leq \\ &\leq p\left(\frac{(n-1)\, s^2}{\sigma^2} \leq c \,\middle|\, H_0\right) = \alpha, \end{aligned}$$

where inequality holds because $\frac{\sigma_0^2}{\sigma^2} \leq 1$, if H_0 is true, and the last equality follows from the distribution of $\frac{(n-1)s^2}{\sigma^2} \sim \chi_{n-1}^2$. The smallest α, which rejects H_0 is called p-value. In this case p-value is equal to the value of the cumulative distribution function of χ_{n-1}^2 distribution at the point $\frac{(n-1)s^2}{\sigma_0^2}$.

The most cost efficient sampling design for estimation of each population parameter is detected by the following procedure:

1. σ_0^2 is computed as min $(\sigma_{mSSRSi}^2, \sigma_{mSSRSh}^2)$.

2. Hypothesis test is done by computing p-value. The p-values are displayed in Figures 4.7 and 4.8 where grey dashed vertical line represent $\alpha = 0.01$.
3. TSSh sampling design is chosen as the most cost efficient sampling design for chosen population parameter and procedure stops if the p-value is less than 0.01. The procedure is continued to the step 4, if the p-value is not less than 0.01.
4. mSSRSi is chosen as the most cost efficient sampling design for chosen population parameter if $\sigma^2_{mSSRSi} < \sigma^2_{mSSRSh}$. mSSRSh is chosen as the most cost efficient sampling design for chosen population parameter if $\sigma^2_{mSSRSi} \geq \sigma^2_{mSSRSh}$.

The expected precision of population parameter estimates by three sampling designs is given in Tables 4.10, 4.11, 4.12 and 4.13. The columns of the tables:

- parameter: the name of population parameter,
- domain: geographical domains (0 "Latvia", 1 "Riga", 2 "Cities (excluding Riga)", 3 "Towns", 4 "Rural areas"),
- age: age group (0 "15–74", 1 "15–14", 2 "15–74"),
- value: the true value of the population parameter,
- sd.mSSRSi: expected standard deviation of the population parameter estimate under the mSSRSi,
- sd.mSSRSh: expected standard deviation of the population parameter estimate under the mSSRSh,
- sd.TSSh: estimated standard deviation of the population parameter estimate under the TSSh,
- p-value: p-value of hypothesis test 4.4,
- des: the most efficient design selected (1 "mSSRSi", 2 "mSSRSh", 3 "TSSh").

The parameters for the whole population and split by age groups are observable in Tables 4.10 and 4.11. The TSSh is the most efficient design in 17 cases from 18. The exception is the estimation of a parameter "total number of employed individuals" where mSSRSi is selected as the most efficient design.

The parameters split additionally by geographical domains are observable in Tables 4.12 and 4.13. The mSSRSi is selected as the most efficient design for two parameters here – "total number of employed individuals" in domains "Riga" and "Cities (excluding Riga)". The mSSRSh is selected as the most efficient design for 10 parameters where all of them represent the population of domains "Riga" and "Cities (excluding Riga)". The TSSh is selected as the most efficient for 24 parameters in domains "Riga" and "Cities (excluding Riga)". TSSh is selected as the most efficient for all 36 parameters in domains "Towns" and "Rural areas".

It is visible that TSSh is definitely the most efficient design for the parameters representing domains "Towns" and "Rural areas". It is because the travelling distances are higher in these domains compared to the domains "Riga" and "Cities (excluding Riga)".

The conclusions are not so straightforward in the case of domains "Riga" and "Cities (excluding Riga)". Each of the three designs has been selected as the most efficient at least for

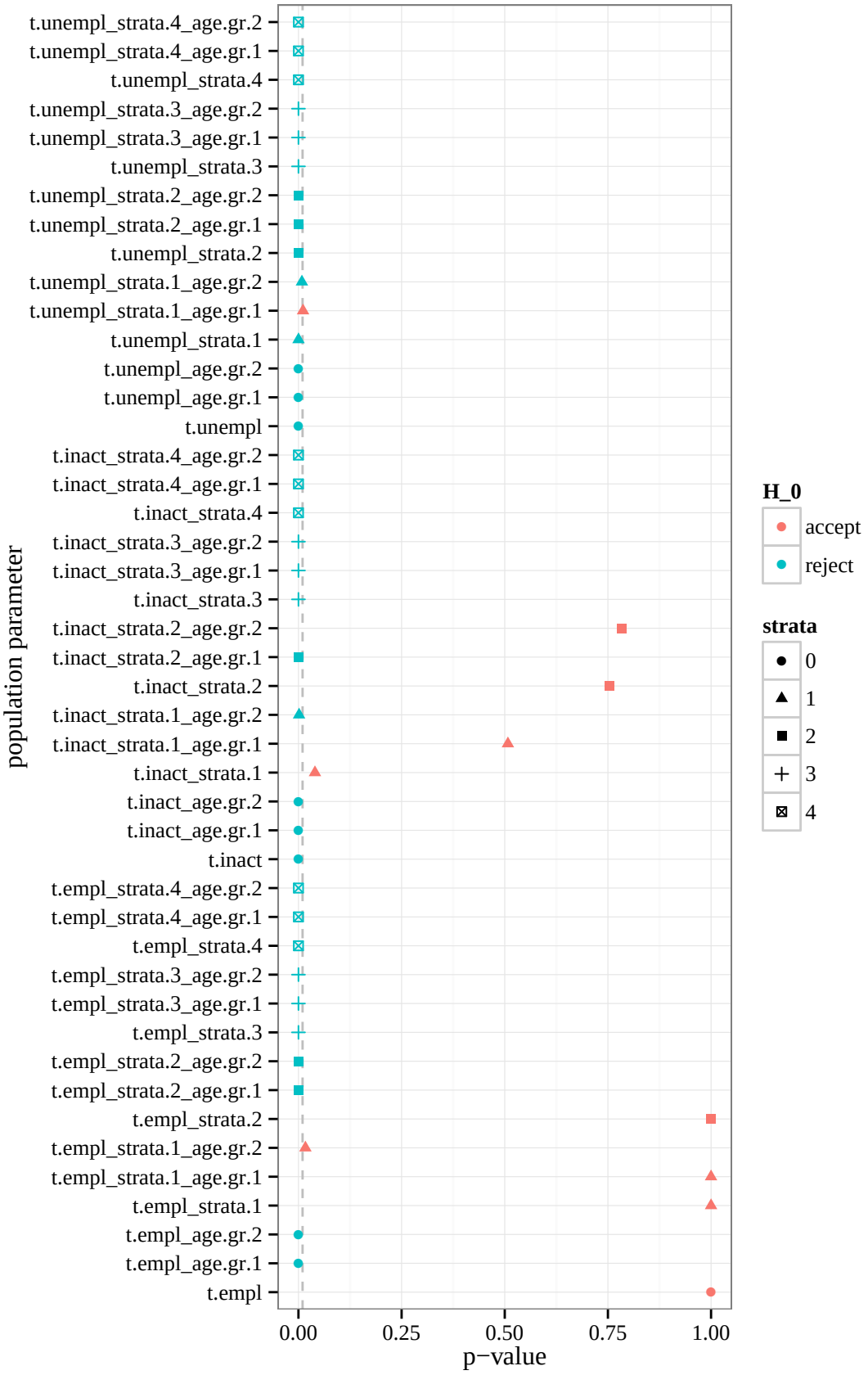


Figure 4.7 **The p-values of hypothesis test for totals**

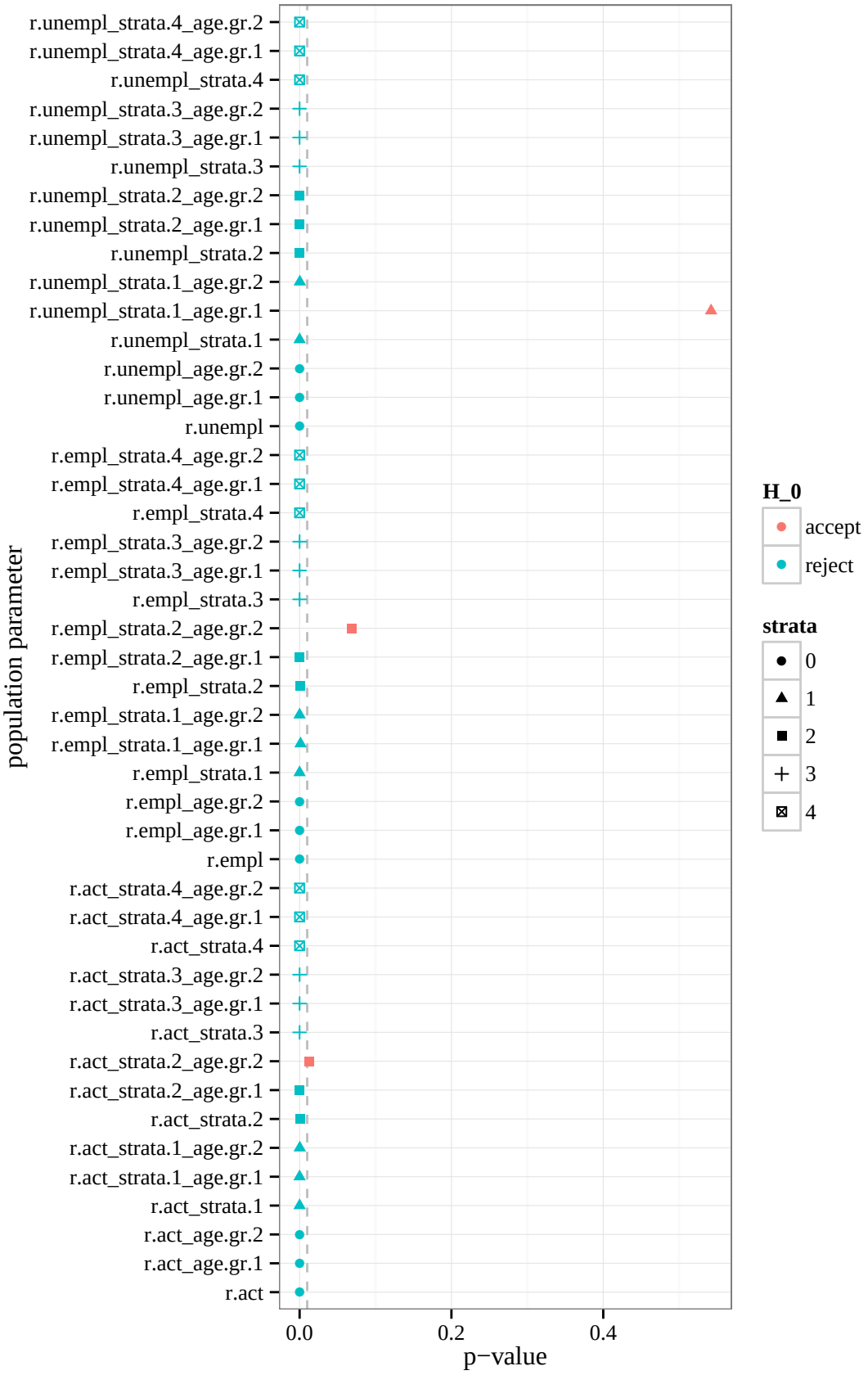


Figure 4.8 **The *p*-values of hypothesis test for ratios**

Table 4.10

The expected precision of total estimates by three sampling designs for domain "Latvia"

parameter	age	value	sd.mSSRSi	sd.mSSRSh	sd.TSSh	p-value	des
t.empl	0	972 327	11 034	12 061	11 437	1.000	1
t.unempl	0	133 746	6 173	4 958	4 654	0.000	3
t.inact	0	545 052	10 513	9 109	8 605	0.000	3
t.empl	1	102 838	5 410	4 344	4 097	0.000	3
t.unempl	1	27 693	2 868	2 191	2 034	0.000	3
t.inact	1	157 176	6 487	5 373	5 078	0.000	3
t.empl	2	869 489	11 204	10 802	10 150	0.000	3
t.unempl	2	106 054	5 565	4 393	4 121	0.000	3
t.inact	2	387 876	9 499	7 800	7 282	0.000	3

some parameters. The TSSh has been determined as the most efficient in 24 cases, mSSRSh in 10 cases and mSSRSi in two cases, though the difference of standard deviations is marginal. The conclusion is that TSSh is selected as the most efficient designs also in domains "Riga" and "Cities (excluding Riga)".

The cost efficiency analysis is done from a conservative position with respect to the TSSh. Firstly, the total sample size of each stratum for the mSSRSi and the mSSRSh is chosen slightly larger compared to the TSSh (Section 4.4). Secondly, the TSSh is chosen as the most efficient design only in the cases when it is supported by a strong evidence (p-value of the hypothesis testing is less than 0.01). The mSSRSi and the mSSRSh are preferred in the cases when there is uncertainty in the determination of the most efficiency design. For example, there are several cases when the precision of estimates achieved by the mSSRSh and the TSSh is quite similar. The TSSh can be used reasonably well in some of these cases even if the mSSRSh has been chosen as the most efficient design, for example, in cases for the estimation of the totals of inactive individuals in the domain "Riga" and the totals of employed individuals aged 25–74 in the domain "Riga" (these are the cases when p-value is slightly higher than 0.01).

The TSSh has achieved the highest precision of estimates in most cases despite the conservative position with respect to it. Therefore it is recommended to use the currently used two-stage sampling design for the Latvian LFS to achieve the highest overall precision under the current budget constrains. Switching to a simpler sampling design will result with one of two negative effects. The first possible negative effect is the loss of overall precision if the survey cost is kept in the current budget level. The second possible negative effect is the increase in survey cost if overall precision level is kept equal to the current level.

Table 4.11

The expected precision of ratio estimates by three sampling designs for domain "Latvia"

parameter	age	value	sd.mSSRSi	sd.mSSRSh	sd.TSSh	p-value	des
r.act	0	0.670	0.00637	0.00485	0.00452	0.000	3
r.empl	0	0.589	0.00668	0.00514	0.00475	0.000	3
r.unempl	0	0.121	0.00546	0.00427	0.00396	0.000	3
r.act	1	0.454	0.01607	0.01219	0.01128	0.000	3
r.empl	1	0.357	0.01550	0.01175	0.01089	0.000	3
r.unempl	1	0.212	0.01967	0.01484	0.01379	0.000	3
r.act	2	0.716	0.00674	0.00533	0.00499	0.000	3
r.empl	2	0.638	0.00721	0.00573	0.00532	0.000	3
r.unempl	2	0.109	0.00557	0.00435	0.00404	0.000	3

Table 4.12

The expected precision of total estimates by three sampling designs and domain

parameter	domain	age	value	sd.mSSRSi	sd.mSSRSh	sd.TSSh	p-value	des
t.empl	1	0	330 855	7 381	8 272	8 329	1.000	1
t.unempl	1	0	47 160	4 284	3 569	3 504	0.000	3
t.inact	1	0	160 949	6 938	6 062	6 009	0.040	2
t.empl	1	1	31 245	3 543	2 903	2 960	1.000	2
t.unempl	1	1	8 152	1 851	1 452	1 435	0.011	2
t.inact	1	1	40 138	3 980	3 300	3 301	0.508	2
t.empl	1	2	299 610	7 533	7 509	7 430	0.017	2
t.unempl	1	2	39 007	3 928	3 222	3 184	0.008	3
t.inact	1	2	120 810	6 322	5 329	5 250	0.001	3
t.empl	2	0	196 200	3 870	4 304	4 126	1.000	1
t.unempl	2	0	26 352	2 125	1 746	1 713	0.000	3
t.inact	2	0	110 307	3 703	3 250	3 261	0.754	2
t.empl	2	1	19 779	1 860	1 532	1 500	0.000	3
t.unempl	2	1	5 362	991	782	764	0.000	3
t.inact	2	1	30 430	2 267	1 903	1 846	0.000	3
t.empl	2	2	176 421	3 926	3 878	3 736	0.000	3
t.unempl	2	2	20 990	1 913	1 536	1 510	0.000	3
t.inact	2	2	79 877	3 360	2 839	2 850	0.784	2
t.empl	3	0	166 623	5 991	6 139	3 325	0.000	3
t.unempl	3	0	23 376	2 493	1 935	1 395	0.000	3
t.inact	3	0	96 256	4 808	4 206	2 549	0.000	3
t.empl	3	1	17 418	2 160	1 687	1 203	0.000	3
t.unempl	3	1	5 101	1 179	873	639	0.000	3
t.inact	3	1	29 682	2 797	2 284	1 593	0.000	3
t.empl	3	2	149 205	5 749	5 487	2 967	0.000	3
t.unempl	3	2	18 275	2 212	1 676	1 224	0.000	3
t.inact	3	2	66 574	4 085	3 361	2 167	0.000	3
t.empl	4	0	278 650	7 004	7 761	5 583	0.000	3
t.unempl	4	0	36 859	3 103	2 405	2 085	0.000	3
t.inact	4	0	177 540	6 129	5 698	4 516	0.000	3
t.empl	4	1	34 396	3 001	2 401	2 043	0.000	3
t.unempl	4	1	9 078	1 568	1 165	1 023	0.000	3
t.inact	4	1	56 926	3 802	3 252	3 013	0.000	3
t.empl	4	2	244 254	6 779	6 787	4 821	0.000	3
t.unempl	4	2	27 781	2 710	2 043	1 754	0.000	3
t.inact	4	2	120 615	5 285	4 461	3 473	0.000	3

Table 4.13

The expected precision of ratio estimates by three sampling designs and domain

parameter	domain	age	value	sd.mSSRSi	sd.mSSRSh	sd.TSSh	p-value	des
r.act	1	0	0.701	0.01287	0.01013	0.00993	0.000	3
r.empl	1	0	0.614	0.01369	0.01091	0.01062	0.000	3
r.unempl	1	0	0.125	0.01110	0.00903	0.00881	0.000	3
r.act	1	1	0.495	0.03661	0.02872	0.02810	0.000	3
r.empl	1	1	0.393	0.03576	0.02807	0.02765	0.001	3
r.unempl	1	1	0.207	0.04215	0.03287	0.03288	0.542	2
r.act	1	2	0.737	0.01341	0.01086	0.01067	0.000	3
r.empl	1	2	0.652	0.01451	0.01187	0.01160	0.000	3
r.unempl	1	2	0.115	0.01133	0.00919	0.00903	0.000	3
r.act	2	0	0.669	0.01113	0.00876	0.00861	0.000	3
r.empl	2	0	0.589	0.01163	0.00920	0.00906	0.001	3
r.unempl	2	0	0.118	0.00934	0.00749	0.00733	0.000	3
r.act	2	1	0.452	0.02879	0.02266	0.02208	0.000	3
r.empl	2	1	0.356	0.02769	0.02175	0.02126	0.000	3
r.unempl	2	1	0.213	0.03524	0.02750	0.02689	0.000	3
r.act	2	2	0.712	0.01173	0.00965	0.00954	0.013	2
r.empl	2	2	0.636	0.01246	0.01026	0.01019	0.069	2
r.unempl	2	2	0.106	0.00946	0.00753	0.00740	0.000	3
r.act	3	0	0.664	0.01462	0.01051	0.00787	0.000	3
r.empl	3	0	0.582	0.01526	0.01105	0.00815	0.000	3
r.unempl	3	0	0.123	0.01248	0.00921	0.00689	0.000	3
r.act	3	1	0.431	0.03589	0.02629	0.01947	0.000	3
r.empl	3	1	0.334	0.03416	0.02493	0.01839	0.000	3
r.unempl	3	1	0.227	0.04618	0.03344	0.02464	0.000	3
r.act	3	2	0.716	0.01544	0.01162	0.00874	0.000	3
r.empl	3	2	0.637	0.01645	0.01247	0.00920	0.000	3
r.unempl	3	2	0.109	0.01261	0.00928	0.00695	0.000	3
r.act	4	0	0.640	0.01132	0.00825	0.00721	0.000	3
r.empl	4	0	0.565	0.01169	0.00857	0.00738	0.000	3
r.unempl	4	0	0.117	0.00947	0.00699	0.00608	0.000	3
r.act	4	1	0.433	0.02589	0.01881	0.01660	0.000	3
r.empl	4	1	0.343	0.02479	0.01798	0.01561	0.000	3
r.unempl	4	1	0.209	0.03228	0.02335	0.02043	0.000	3
r.act	4	2	0.693	0.01219	0.00927	0.00813	0.000	3
r.empl	4	2	0.622	0.01281	0.00977	0.00844	0.000	3
r.unempl	4	2	0.102	0.00961	0.00707	0.00611	0.000	3

Main Results

The aim of this thesis was to develop a framework for the analysis of the cost efficiency of sampling designs. The study started with an in-depth analysis of the two-stage sampling design used for the Latvian Labour Force Survey (LFS). The study continued with the creation of artificial population data representing the target population of the Latvian LFS at the individual level. The final part of the study was the development of a framework for the analysis of the cost efficiency of sampling designs. The main results of thesis are published in three scientific publications (Liberts, 2010a, 2010b, 2013a).

The main goal of this thesis has been achieved and the following results have been obtained:

1. The sampling frame of the primary sampling units (areas) was updated. The updated sampling frame of areas is used for several sample surveys run by the Central Statistical Bureau of Latvia. The update of the frame reduced significantly the coverage errors of the frame.
2. The redesign of the sampling design used for the LFS, the Household Budget Survey and the Survey of Domestic Travellers has been done in the scope of this thesis. The new design has been successfully implemented and has been in use since 2010.
3. A methodology for generating artificial population data has been developed. The methodology allows for the generation of artificial population data close to the real population data, according to the dimension and statistical properties of the real population. Both static and dynamic artificial population data can be created using this methodology.
4. The artificial population data has been created using the developed methodology. The data from the Statistical Household Register and the LFS were used as the input data. The data created are similar (in statistical properties) to the LFS target population data. The artificial population data are dynamic. The changes over time in the artificial population data are similar to the changes over time observed in the LFS. The artificial population data are used in the Monte Carlo simulation experiments carried out in the analysis.
5. Modified stratified simple random sampling design (mSSRS), ensuring even sample allocation by weeks, is introduced as an alternative to the currently used two-stage sampling design. The mSSRS allows sampling of a unit not more then once in a defined time period. The variance formula for the π-estimator of a population total and the approximate variance formula for the π-estimator of a ratio are derived. The design can be used to sample individuals or households.

6. The framework for the analysis of sampling designs with the respect to cost efficiency has been developed. The framework is based on analytical methods and Monte Carlo simulation experiments. The framework allows the user to gain information about the sampling design properties (for example, expected fieldwork cost, expected precision) in a relatively short time and with relatively low cost. This information is very valuable information for the survey planning and decision making process. The advantage of the framework is that no extra data collection is required. The framework utilises data already available to a statistical institute (administrative records, population census data or sample survey data).
7. The set of procedures is developed to support the implementation of the framework in practice. The procedures are developed in R, which is a free software environment for statistical computing and graphics. The procedures are used for Monte Carlo simulations of sampling designs. The procedures are modular – it allows for the extension of the set with additional procedures. There are no limitations on the types of design that can be analysed by the procedures. The only requirement is that it must be possible to write the sampling process of the sampling design under analyses as an R function.
8. The cost efficiency of three sampling designs is estimated using the developed framework. The properties of the chosen sampling designs are explored and recommendations regarding the appropriate sampling design for the LFS are given.
9. It is proven that the two-stage sampling design used currently for the LFS, when compared to two other sampling designs, provides more precise population parameter estimates under the condition of fixed fieldwork cost.

Acknowledgements

First and foremost I offer my gratitude to my thesis consultant Jānis Lapiņš for his guidance, advice, and encouragement he was providing throughout my doctoral studies. I have learned a lot from him during countless discussions we have had. I express my greatest gratitudes also to my thesis supervisor Aleksandrs Šostaks for his support and advice I have received during the studies.

I express my gratitudes to my colleagues at the Central Statistical Bureau, especially to Aija Žīgure and Ieva Aināre, for the support I have received during my doctoral studies. I am also thankful to Gunnar Kulldorff and Imbi Traat for organising my study visits at the University of Umeå and the University of Tartu.

I express my thanks to Rebecca Gillard for helping to improve the language of the thesis. Any remaining grammatical mistakes are my own. I am very thankful to the Stack Exchange community for the knowledge I have gained by searching and browsing the forums like "Stack Overflow"[1], "Cross Validated"[2], "TeX – LaTeX Stack Exchange"[3], and "English Language & Usage Stack Exchange"[4]. I am especially thankful for the promptly and suitable answers I have received to my questions asked at the forums mentioned.

This work has been supported by the European Social Fund within the project «Support for Doctoral Studies at University of Latvia – 2». I am thankful also to the Linda Peetre Memorial Fund for the scholarship I have received in 2011.

Finally I offer my warmest gratitudes to the family, especially to Inga, for patience, understanding and support I have received.

[1] `http://stackoverflow.com/`
[2] `http://stats.stackexchange.com/`
[3] `http://tex.stackexchange.com/`
[4] `http://english.stackexchange.com/`

References

Anderson, T. W., & Darling, D. A. (1952). A test of goodness-of-fit. *Journal of the American Statitical Association*, *49*, 765–769.

Beardwood, J., Halton, J. H., & Hammersley, J. M. (1959). The shortest path through many points. *Mathematical Proceedings of the Cambridge Philosophical Society*, *55*, 299-327.

Calinescu, M., Bhulai, S., & Schouten, B. (2013). Optimal resource allocation in survey designs. *European Journal of Operational Research*, *226*(1), 115–121.

Carkova, V. (2001). *Markova ķēdes* [Markov chains] (Study aid). Riga: University of Latvia.

Central Statistical Bureau of Latvia. (2012a). *Employment and unemployment* [Metadata]. Riga. Retrieved 15.12.2012, from `http://ej.uz/CSB-LFS`

Central Statistical Bureau of Latvia. (2012b). *PCG03. average retail prices of selected commodity* [Table]. Riga. Retrieved 23.09.2012, from `http://ej.uz/CSB-PCG03`

Chen, B.-C. (2008). *Stochastic simulation of field operations in surveys* (Research Rep.) Washington: U. S. Census Bureau. Retrieved from `https://www.census.gov/srd/www/byyear.html`

Cox, L. (2012). The case for simulation models of federal surveys. In *Research conference papers of federal committee on statistical methodology research conference 2012.* Washington. Retrieved from `http://www.fcsm.gov/events/papers2012.html`

Dowle, M., Short, T., & Lianoglou, S. (2012). data.table: Extension of data.frame for fast indexing, fast ordered joins, fast assignment, fast grouping and list columns [Computer software]. (R package version 1.8.6)

European Commission. (2012a). *Labour force survey in the EU, candidate and EFTA countries – Main characteristics of national surveys, 2011* (Tech. Rep.). Luxembourg: Eurostat. Retrieved from `http://epp.eurostat.ec.europa.eu/`

European Commission. (2012b). *Quality report of the European Union Labour Force Survey – 2010* (Tech. Rep.). Luxembourg: Eurostat. Retrieved from `http://epp.eurostat.ec.europa.eu/`

Ģeodēziskās atskaites sistēmas un topogrāfisko karšu sistēmas noteikumi. (2011). Latvijas Vēstnesis. Retrieved from `http://www.likumi.lv/doc.php?id=239759` (in Latvian)

Groves, R. M. (1989). *Survey errors and survey costs*. New Jersey: Wiley.

Hahsler, M., & Hornik, K. (2007). TSP–infrastructure for the traveling salesperson problem. *Journal of Statistical Software*, *23*(2), 1–21. Retrieved from http://www.jstatsoft.org/v23/i02

Hahsler, M., & Hornik, K. (2011). Traveling salesperson problem (TSP) [Computer software]. (R package version 1.0-7.)

Hansen, M. H., Hurwitz, W. N., & Madow, W. G. (1953). *Sample survey methods and theory* (Vol. I). New-York: Wiley.

Heeringa, S., & Groves, R. (2006). Responsive design for household surveys: tools for actively controlling survey errors and costs. *Journal of the Royal Statistical Society: Series A (Statistics in Society)*, *169*(3), 439–457.

Jessen, R. J. (1942). *Statistical investigation of a sample survey for obtaining farm facts* (Research Bulletin No. 304). Iowa State College of Agriculture and Mechanic Arts.

Kalsbeek, W., Botman, S., Massey, J., & Liu, P.-W. (1994). Cost-efficiency and the number of allowable call attempts in the national health interview survey. *Journal of Official Statistics*, *10*(2), 133–152.

Kane, M. J., & Emerson, J. W. (2012a). bigmemory: Manage massive matrices with shared memory and memory-mapped files [Computer software]. (R package version 4.3.0)

Kane, M. J., & Emerson, J. W. (2012b). bigtabulate: table-, tapply-, and split-like functionality for matrix and big.matrix objects [Computer software]. (R package version 1.1.0)

Kish, L. (1965). *Survey sampling*. New-York: John Wiley & Sons.

Lapiņš, J. (1997). Sampling surveys in Latvia: Current situation, problems and future development. *Statistics in Transition*, *3*(2), 281–292.

Lapiņš, J., Vaskis, E., Priede, Z., & Bāliņa, S. (2002). Household surveys in Latvia. *Statistics in Transition*, *5*(4), 617–641. Retrieved from http://www.stat.gov.pl/pts/15_ENG_HTML.htm

Liberts, M. (2010a). The redesign of Latvian Labour Force Survey. In M. Carlson, H. Nyquist, & M. Villani (Eds.), *Official statistics – methodology and applications in honour of Daniel Thorburn* (pp. 193–203). Stockholm, Sweden: Stockholm University. Retrieved from http://officialstatistics.wordpress.com/

Liberts, M. (2010b). The weighting in household sample surveys. In O. Krastiņš & I. Vanags (Eds.), *The results of statistical scientific research 2010* (pp. 168–174). Riga: Central Statistical Bureau of Latvia. Retrieved from http://home.lu.lv/~pm90015/work/PhD/pub/10Papers/Liberts_2010_Weighting.pdf

Liberts, M. (2013a). *The cost efficiency of sampling designs*. Manuscript submitted for publication in the journal *Statistics in Transition – new series*.

Liberts, M. (2013b). Survey-design-simulation [Computer program]. Published online. Retrieved from https://github.com/djhurio/Survey-Design-Simulation

Lilliefors, H. (1967). On the Kolmogorov-Smirnov test for normality with mean and variance unknown. *Journal of the American Statitical Association*, *62*, 399–402.

Mahalanobis, P. C. (1940). A sample survey of the acreage under jute in Bengal. *Sankhyā: The Indian Journal of Statistics*, *4*(4), 511–530.

Malec, D. (1995). Selecting multiple-objective fixed-cost sample designs using an admissibility criterion. *Journal of Statistical Planning and Inference*, *48*(2), 229–240.

Mohl, C., & Laflamme, F. (2007). Research and responsive design options for survey data collection at statistics Canada. In *Proceedings of survey research methods section at joint statistical meeting 2007.*

Mood, A. M., Graybill, F. A., & Boes, D. C. (1974). *Introduction to the theory of statistics*. McGraw-Hill Book Company.

Pinheiro, J., Bates, D., DebRoy, S., Sarkar, D., & R Core Team. (2012). nlme: Linear and nonlinear mixed effects models [Computer software]. (R package version 3.1-105)

Population register law. (1998). Latvijas Vēstnesis. Retrieved from `http://www.likumi.lv/doc.php?id=49641` (in Latvian)

R Core Team. (2013). R: A language and environment for statistical computing [Computer software]. Vienna, Austria. Retrieved from `http://www.r-project.org`

Revolution Analytics. (2012a). doMC: Foreach parallel adaptor for the multicore package [Computer software]. (R package version 1.2.5)

Revolution Analytics. (2012b). foreach: Foreach looping construct for R [Computer software]. (R package version 1.4.0)

Robert, C. P., & Casella, G. (2004). *Monte Carlo statistical methods*. Springer.

Robert, C. P., & Casella, G. (2010). *Introducing Monte Carlo methods with R*. Springer.

Rosenkrantz, D., Stearns, R., & Lewis, P., II. (1977). An analysis of several heuristics for the traveling salesman problem. *SIAM Journal on Computing*, *6*(3), 563–581.

Särndal, C.-E., Swensson, B., & Wretman, J. (1992). *Model assisted survey sampling*. New-York: Springer.

United Nations. (2010). *Handbook on population and housing census editing: Revision 1*. New York: United Nations.

Wasserman, L. (2004). *All of statistics*. New-York: Springer.

Wasserman, L. (2006). *All of nonparametric statistics*. New-York: Springer.

Wolter, K. M. (2007). *Introduction to variance estimation*. New-York: Springer.

Appendix 1

R Functions for Monte Carlo Simulations

1.1 Sample generation functions

1.1.1 SRS

The name of procedure is `SamplingSRS`. The aims of the procedure are:

- Drawing a simple random sample without replacement from a given population frame according to Särndal et al. (1992, p. 66).
- Computing design weights for the sampled units according to the inclusion probabilities (Särndal et al., 1992, p. 30).

The arguments of the procedure:

- `frame.1`: population frame with rows as sampling units and columns as variables.
- `n`: sample size of units.
- `name.weight`: the name of a weight variable.

The steps of the procedure:

- The package `bigmemory` is loaded. It is necessary to manipulate arguments with the `big.matrix` class (Kane & Emerson, 2012a).
- Converting class of the arguments if necessary.
- Population size N is computed as the number of rows of `frame.1`.
- Testing the coherence of the arguments.
- The data frame `s.1` is created by selecting randomly n rows from `frame.1` without replacement. The function `sample` is used for a random selection (R Core Team, 2013). The sampled units are randomly sorted.
- Weight variable with name `name.weight` is computed as $\frac{N}{n}$.
- Weight variable is attached to `s.1`.
- `s.1` is returned as an output of the procedure.

```
1  SamplingSRS <- function(frame.1, n = 30, name.weight = ".dw") {
2
3    # Libs
4    require(bigmemory)
```

```
5
6     # Argument class conversation
7
8     if (!is.big.matrix(frame.1)) frame.1 <- as.data.frame(frame.1)
9     n <- as.integer(n[1])
10    name.weight <- as.character(name.weight[1])
11
12    # Testing
13
14    N <- nrow(frame.1)
15    if (n<=0 | n>N) stop("n has to be in 0-N")
16    if (name.weight %in% colnames(frame.1))
17      print("WARNING: Weight variable exists, it will be overwritten")
18
19    # Sampling
20
21    s.1 <- as.data.frame(frame.1[sample(N, n), ])
22    s.1[name.weight] <- N/n
23
24    return(s.1)
25  }
```

1.1.2 SRS by Weeks

The name of procedure is `SamplingSRSWeek`. The aims of the procedure are:

- Drawing a simple random sample without replacement from a given population frame according to Särndal et al. (1992, p. 66).
- Partitioning the sample in number of random sub-samples with equal number of units in each sub-sample. The sub-samples are distributed by weeks.
- Computing design weights for the sampled units according to the inclusion probabilities (Särndal et al., 1992, p. 30).

The arguments of the procedure:

- `frame.1`: population frame with rows as sampling units and columns as variables.
- `n`: sample size of units.
- `name.weight`: the name of a weight variable.
- `name.week`: the name of a week variable.
- `weeks`: the number of weeks denoted by W.

The steps of the procedure:

- The package `bigmemory` is loaded. It is necessary to manipulate arguments with the `big.matrix` class (Kane & Emerson, 2012a).
- Converting class of the arguments if necessary.
- Population size N is computed as the number of rows of `frame.1`.
- Sample size n is rounded to the closest multiple of W.
- Testing the coherence of the arguments.
- The data frame `s.1` is created by selecting randomly n rows from `frame.1` without replacement. The function `sample` is used for a random selection (R Core Team, 2013). The sampled units are randomly sorted.

- Weight variable with the name `name.weight` is computed as $\frac{N}{n}$ and attached to `s.1`.
- Week variable with the name `name.week` is computed by the procedure: the number 1 is assigned to the first $\frac{n}{W}$ sampled units, the number 2 is assigned to the next $\frac{n}{W}$ sampled units and so on till the number W is assigned to the last $\frac{n}{W}$ sampled units.
- Week variable is attached to `s.1`.
- `s.1` is returned as an output of the procedure.

```
SamplingSRSWeek <- function(frame.1,
                            n = 30,
                            name.weight = ".dw",
                            name.week = ".week",
                            weeks = 1) {

  # Libs
  require(bigmemory)

  # Argument type convertion

  if (!is.big.matrix(frame.1)) frame.1 <- as.data.frame(frame.1)
  n <- as.integer(n)[1]
  name.weight <- as.character(name.weight)[1]
  name.week <- as.character(name.week)[1]
  weeks <- as.integer(weeks)[1]

  # Testing

  if (weeks < 1) weeks <- 1

  N <- nrow(frame.1)
  n <- round(n / weeks) * weeks

  if (n < weeks) n <- weeks
  if (n > N) n <- N %/% weeks * weeks

  if (n %% weeks > 0) stop("n is not a multiple of weeks")

  if (name.weight %in% colnames(frame.1))
    print("WARNING: Weight variable exists, it will be overwritten")
  if (name.week %in% colnames(frame.1))
    print("WARNING: Week variable exists, it will be overwritten")

  # Sampling

  s.1 <- data.frame(frame.1[sample(N, n), ])
  s.1[name.weight] <- N/n
  s.1[name.week] <- rep(1:weeks, each = n / weeks)
  return(s.1)
}
```

1.1.3 Cluster Sampling

The name of procedure is `SamplingCluster`. The aims of the procedure are:

- Drawing a simple random cluster sample without replacement from a given population frame according to Särndal et al. (1992, p. 129)
- Computing design weights for the sampled units according to the inclusion probabilities (Särndal et al., 1992, p. 30)

The arguments of the procedure:

- `frame.1`: population frame with rows as sampling units and columns as variables
- `frame.2`: population frame with rows as clusters of units and columns as variables
- `n`: sample size of clusters
- `name.weight`: the name of a weight variable
- `name.cluster`: the name of a cluster variable in `frame.1` and `frame.2`

The steps of the procedure:

- The package `bigmemory` is loaded. It is necessary to manipulate arguments with the `big.matrix` class (Kane & Emerson, 2012a).
- Converting class of the arguments if necessary.
- Population size N is computed as the number of rows of `frame.2`.
- Testing the coherence of the arguments.
- n clusters are selected randomly from the `frame.2` without replacement. The function `sample` is used for a random selection (R Core Team, 2013). The sampled clusters are randomly sorted.
- The data frame `s.1` is created by selecting sampling units belonging to sampled clusters.
- Weight variable with the name `name.weight` is computed as $\frac{N}{n}$ and attached to `s.1`.
- `s.1` is returned as an output of the procedure.

```
SamplingCluster <- function(frame.1, frame.2, n=30, name.weight=".dw",
                            name.cluster) {

  # Libs
  require(bigmemory)

  # Argument type convertion

  if (!is.big.matrix(frame.1)) frame.1 <- as.data.frame(frame.1)
  if (!is.big.matrix(frame.2)) frame.2 <- as.data.frame(frame.2)
  n <- as.integer(n)
  name.weight <- as.character(name.weight)
  name.cluster <- as.character(name.cluster)

  # Testing

  N <- nrow(frame.2)

  if (n <= 0 | n > N) stop("ERROR: n has to be in 0-N")
  if (name.weight %in% colnames(frame.1))
    print("WARNING: Weight variable exists, it will be overwrited")
  if (!name.cluster %in% colnames(frame.1))
    stop("ERROR: Can not find cluster variable in frame.1")
  if (!name.cluster %in% colnames(frame.2))
    stop("ERROR: Can not find cluster variable in frame.2")

  # Sampling

  s.2 <- as.vector(frame.2[, name.cluster][sample(N, n)])

  s.1 <- as.data.frame(frame.1[frame.1[,name.cluster] %in% s.2, ])
  s.1[name.weight] <- N/n

```

```
  return(s.1)
}
```

1.1.4 Cluster Sampling by Weeks

The name of procedure is `SamplingClusterWeek`. The aims of the procedure are:

- Drawing a simple random cluster sample without replacement from a given population frame according to Särndal et al. (1992, p. 129).
- Partitioning the sampled clusters in number of random sub-samples with equal number of clusters in each sub-sample. The sub-samples are distributed by weeks.
- Computing design weights for the sampled units according to the inclusion probabilities (Särndal et al., 1992, p. 30).

The arguments of the procedure:

- `frame.1`: population frame with rows as sampling units and columns as variables
- `frame.2`: population frame with rows as clusters of units and columns as variables
- `n`: sample size of units
- `name.weight`: the name of a weight variable
- `name.cluster`: the name of a cluster variable in `frame.1` and `frame.2`
- `name.week`: the name of a week variable
- `weeks`: the number of weeks denoted by W

The steps of the procedure:

- The package `bigmemory` is loaded. It is necessary to manipulate arguments with the `big.matrix` class (Kane & Emerson, 2012a).
- Converting class of the arguments if necessary.
- Population size N is computed as the number of rows of `frame.2`.
- Sample size n is rounded to the closest multiple of W.
- Testing the coherence of the arguments.
- n clusters are selected randomly from the `frame.2` without replacement. The function `sample` is used for a random selection (R Core Team, 2013). The sampled clusters are randomly sorted.
- Week variable with the name `name.week` is computed by the procedure: the number 1 is assigned to the first $\frac{n}{W}$ sampled clusters, the number 2 is assigned to the next $\frac{n}{W}$ sampled clusters and so on till the number W is assigned to the last $\frac{n}{W}$ sampled clusters.
- The data frame `s.1` is created from units belonging to the sampled clusters.
- Weight variable with the name `name.weight` is computed as $\frac{N}{n}$ and attached to `s.1`.
- Week variable is attached to `s.1`.
- `s.1` is returned as an output of the procedure.

```
SamplingClusterWeek <- function(frame.1,
                                frame.2,
                                n=30,
```

```
                                    name.weight=".dw",
                                    name.cluster,
                                    name.week = ".week",
                                    weeks = 1) {

  # Libs
  require(bigmemory)

  # Argument type convertion

  if (!is.big.matrix(frame.1)) frame.1 <- as.data.frame(frame.1)
  if (!is.big.matrix(frame.2)) frame.2 <- as.data.frame(frame.2)
  n <- as.integer(n)[1]
  name.weight <- as.character(name.weight)[1]
  name.cluster <- as.character(name.cluster)[1]
  name.week <- as.character(name.week[1])
  weeks <- as.integer(weeks[1])

  # Testing

  if (weeks < 1) weeks <- 1

  N <- nrow(frame.2)
  n <- round(n / weeks) * weeks

  if (n < weeks) n <- weeks
  if (n > N) n <- N %/% weeks * weeks

  if (n %% weeks > 0) stop("n is not a multiple of weeks")

  if (name.weight %in% colnames(frame.1))
    print("WARNING: Weight variable exists, it will be overwritten")
  if (name.week %in% colnames(frame.1))
    print("WARNING: Week variable exists, it will be overwritten")

  if (!name.cluster %in% colnames(frame.1))
    stop("ERROR: Can not find cluster variable in frame.1")
  if (!name.cluster %in% colnames(frame.2))
    stop("ERROR: Can not find cluster variable in frame.2")

  # Sampling

  s.2 <- as.vector(frame.2[, name.cluster][sample(N, n)])

  tmp <- data.frame(s.2, rep(1:weeks, each = n / weeks))
  names(tmp) <- c(name.cluster, name.week)

  s.1 <- data.frame(frame.1[frame.1[, name.cluster] %in% s.2, ])
  s.1[name.weight] <- N/n
  s.1[name.week] <- NULL
  s.1 <- merge(s.1, tmp, by = name.cluster, sort = F)
  s.1 <- s.1[c(colnames(frame.1), name.weight, name.week)]

  return(s.1)
}
```

1.1.5 Stratified Cluster Sampling

The name of procedure is `SamplingClusterStr`. The aims of the procedure are:

- Drawing a simple random stratified cluster sample without replacement from a given population frame according to Särndal et al. (1992, p. 100, 129)
- Computing design weights for the sampled units according to the inclusion probabilities (Särndal et al., 1992, p. 30)

The arguments of the procedure:

- `frame.1`: population frame with rows as sampling units and columns as variables
- `frame.2`: population frame with rows as clusters of units and columns as variables
- `n`: the vector of sample sizes of clusters by strata
- `name.weight`: the name of a weight variable
- `name.cluster`: the name of a cluster variable in `frame.1` and `frame.2`
- `name.strata`: the name of a stratification variable in `frame.2`

The steps of the procedure:

- The package `bigmemory` is loaded. It is necessary to manipulate arguments with the `big.matrix` class (Kane & Emerson, 2012a).
- The package `bigtabulate` is loaded. The function `bigtable` will be used from the package (Kane & Emerson, 2012b).
- Converting class of the arguments if necessary.
- Population size $\boldsymbol{N}$ is computed as vector with size equal to the number of strata in `frame.2` where N_h is the number of clusters in stratum h of `frame.2`.
- Testing the coherence of the arguments.
- n_h clusters are selected randomly from the stratum h of `frame.2` without replacement. The function `sample` is used for a random selection in each stratum (R Core Team, 2013).
- The data frame `s.1` is created by selecting sampling units belonging to sampled clusters.
- Weight variable with the name `name.weight` is computed as $\frac{N_h}{n_h}$ and attached to `s.1`.
- `s.1` is returned as an output of the procedure.

```
SamplingClusterStr <- function(frame.1,
                               frame.2,
                               n,
                               name.weight=".dw",
                               name.cluster,
                               name.strata) {

  ### Libs
  require(bigmemory)
  require(bigtabulate)

  ### Argument type convertion
  if (!is.big.matrix(frame.1)) frame.1 <- data.frame(frame.1)
  if (!is.big.matrix(frame.2)) frame.2 <- data.frame(frame.2)
  n <- as.vector(as.integer(n))
  name.weight <- as.character(name.weight)[1]
  name.cluster <- as.character(name.cluster)[1]
  name.strata <- as.character(name.strata)[1]

  ### Testing
  if (name.weight %in% colnames(frame.1))
    print("WARNING: Weight variable exists in frame.1, it will be overwrited")
```

```
  if (!name.cluster %in% colnames(frame.1))
    stop("ERROR: Can not find cluster variable in frame.1")
  if (!name.cluster %in% colnames(frame.2))
    stop("ERROR: Can not find cluster variable in frame.2")
  if (!name.strata %in% colnames(frame.2))
    stop("ERROR: Can not find strata variable in frame.2")
  if (any(frame.2[, name.strata] != sort(frame.2[, name.strata])))
    stop("ERROR: frame.2 is not sorted by strata")

  N <- as.vector(bigtable(frame.2, name.strata))
  a <- cumsum(N) - N

  if (any(n < 0)) stop("ERROR: n has to be greater than 0")

  # Reduction of n, if n>N
  n <- ifelse(n>N, N, n)

  if (sum(n) == 0) stop("Total sample size is 0")

  N <- N[n>0]
  a <- a[n>0]
  n <- n[n>0]

  ### Sampling
  s.2.index <- unlist(mapply(sample.mod, N, n, a, SIMPLIFY = F))
  s.2 <- frame.2[s.2.index, name.cluster]
  s.1 <- data.frame(frame.1[frame.1[, name.cluster] %in% s.2, ])

  ### Weighting
  w <- N / n
  s.1[name.weight] <- w[match(s.1[, name.strata],
                              sort(unique(s.1[, name.strata])))]

  return(s.1)
}
```

1.1.6 Two-Stage Sampling

The name of procedure is `SamplingTwoStage`. The aims of the procedure are:

- Drawing a two-stage sample from a given population frame according to the sampling design defined in Chapter 1.
- Computing design weights for the sampled units according to the inclusion probabilities (Särndal et al., 1992, p. 30)

The arguments of the procedure:

- `frame.PSU`: the name of a population frame with rows as primary sampling units (PSU) and columns as variables.
- `frame.SSU`: the name of a population frame with rows as secondary sampling units (SSU) and columns as variables.
- `frame.TSU`: the name of a population frame with rows as tertiary sampling units (TSU) and columns as variables.
- `name.weight.s1`: the name of a weight variable for PSU design weights.

- `name.weight.s2`: the name of a weight variable for conditional SSU and TSU design weights.
- `name.weight`: the name of a weight variable for ultimate SSU and TSU design weights.
- `name.week`: the name of a week variable.
- `name.PSU`: the name of a PSU variable in `frame.PSU`, `frame.SSU` and `frame.TSU`.
- `name.SSU`: the name of a SSU variable in `frame.SSU` and `frame.TSU`.
- `name.strata`: the name of a stratification variable in `frame.2`.
- `param`: matrix of design parameters with a row per stratum and nine columns defining nine design parameters for each stratum. See Table 1.1 for example of `param`. There should be the columns named as `s`, `A`, `B`, `W`, `d`, `Q`, `w`, `M`, `m` in the `param`, where:
  - `s` is the label of strata.
  - `A` is the number of the vertices of polygon.
  - `B` is the number of polygons.
  - `W` is the period of the rotation scheme defined by number of weeks.
  - `d` is the sampling displacement δ_h according to Section 1.3.2.
  - `Q` is the rotation speed q_h according to Section 1.3.3.
  - `w` is the length of sample to be drawn defined by number of weeks.
  - `M` is the number of SSUs in the population frame `frame.SSU`.
  - `m` is the number of SSUs to be sampled in each sampled PSU.

The steps of the procedure:

- The package `bigmemory` is loaded. It is necessary to manipulate arguments with the `big.matrix` class (Kane & Emerson, 2012a).
- The package `bigtabulate` is loaded. The function `bigtable` will be used from the package (Kane & Emerson, 2012b).
- Converting class of the arguments if necessary.
- Testing the coherence of the arguments.
- Random values ξ_h are computed according to Section 1.3.2.
- Sampling points $a_{j,h,k,i}$ are computed by (1.3).
- A PSU sample is selected according the $a_{j,h,k,i}$.
- PSU design weights with the name `name.weight.s1` are computed according Liberts (2010b, p. 171).
- A TSU sample is selected by means of stratified cluster sampling using the procedure `SamplingClusterStr` in sampled PSUs. PSUs are used as strata and SSUs are used as clusters here. TSU sample with conditional design weights with name `name.weight.s2` and computed according Liberts (2010b, p. 172) are achieved.
- Ultimate SSU and TSU design weights with a name `name.weight` are computed according Liberts (2010b, p. 172).
- Week variable with a name `name.week` is attached to the TSU sample.
- TSU sample is returned as an output of the procedure.

Table 1.1

Example of `param`

s	A	B	W	d	Q	w	M	m
1	8	1	13	0.002877	0	13	256556	10
2	8	2	13	0.004115	0	13	157709	7
3	8	2	13	0.003690	0	13	129823	8
4	8	2	13	0.003582	0	13	198515	9

```
SamplingTwoStage <- function(frame.PSU,
                             frame.SSU,
                             frame.TSU,
                             name.weight.s1 = ".dw1",
                             name.weight.s2 = ".dw2",
                             name.weight = ".dw",
                             name.week = "week",
                             name.PSU,
                             name.SSU,
                             name.strata,
                             param) {

  ### Libs
  require(bigmemory)
  require(bigtabulate)

  ### Argument type convertion
  if (!is.big.matrix(frame.PSU)) frame.PSU <- data.frame(frame.PSU)
  if (!is.big.matrix(frame.SSU)) frame.SSU <- data.frame(frame.SSU)
  if (!is.big.matrix(frame.TSU)) frame.TSU <- data.frame(frame.TSU)

  name.weight.s1 <- as.character(name.weight.s1[1])
  name.weight.s2 <- as.character(name.weight.s2[1])
  name.weight <- as.character(name.weight[1])

  name.PSU <- as.character(name.PSU[1])
  name.SSU <- as.character(name.SSU[1])

  name.strata <- as.character(name.strata[1])

  param <- data.frame(param)


  ### Testing
  if (name.weight %in% colnames(frame.PSU)) {
    print("WARNING: Weight variable exists in frame.PSU")
    print("WARNING: It will be overwrited")
  }
  if (name.weight %in% colnames(frame.SSU)) {
    print("WARNING: Weight variable exists in frame.SSU")
    print("WARNING: It will be overwrited")
  }
  if (name.weight %in% colnames(frame.TSU)) {
    print("WARNING: Weight variable exists in frame.TSU")
    print("WARNING: It will be overwrited")
  }

  if (!name.PSU %in% colnames(frame.PSU))
    stop("ERROR: Can not find PSU variable in frame.PSU")
  if (!name.PSU %in% colnames(frame.SSU))
```

```
  stop("ERROR: Can not find PSU variable in frame.SSU")
if (!name.PSU %in% colnames(frame.TSU))
  stop("ERROR: Can not find PSU variable in frame.TSU")

if (!name.SSU %in% colnames(frame.SSU))
  stop("ERROR: Can not find SSU variable in frame.SSU")
if (!name.SSU %in% colnames(frame.TSU))
  stop("ERROR: Can not find SSU variable in frame.TSU")

if (!name.strata %in% colnames(frame.PSU))
  stop("ERROR: Can not find strata variable in frame.PSU")

if (any(frame.PSU[, name.strata] != sort(frame.PSU[, name.strata])))
  stop("ERROR: frame.PSU is not sorted by strata")
if (any(frame.PSU[, name.PSU] != sort(frame.PSU[, name.PSU])))
  stop("ERROR: frame.PSU is not sorted by PSU")

if (any(frame.SSU[, name.strata] != sort(frame.SSU[, name.strata])))
  stop("ERROR: frame.SSU is not sorted by strata")
if (any(frame.SSU[, name.PSU] != sort(frame.SSU[, name.PSU])))
  stop("ERROR: frame.SSU is not sorted by PSU")


# Function nemesis
nemesis <- function(s, A, b, W, d, Q, w, M, hi, add) {

  sample.step <- Q / W + (1 + d) / A / W

  sample.PSU <- data.frame(strata = s,
                           b = b,
                           i = rep(1:A, w),
                           week = rep(1:w, each=A))

  sample.PSU$a <- (hi +
    (b - 1) * (1 + d) * (A + 1) / (A * b) +
    (sample.PSU$i - 1) * (1 + d) / A +
    (sample.PSU$week - 1) * sample.step) %% 1

  sample.PSU$x <- sin(2 * pi * sample.PSU$a) / 2 / pi
  sample.PSU$y <- cos(2 * pi * sample.PSU$a) / 2 / pi

  sample.PSU$A <- sample.PSU$a * M + add

  return(sample.PSU)
}


# Random number
param$hi <- runif(nrow(param))

# Add
param$add <- cumsum(param$M) - param$M

# Function for param.2
fun.tmp1 <- function(b) data.frame(param[param$B >= b, ], b=b)

param.2 <- do.call(rbind, lapply(1:max(param$B), fun.tmp1))
param.2 <- param.2[order(param.2$s, param.2$b), ]
rownames(param.2) <- NULL

# Sampling
s.1 <- mapply(nemesis,
              s = param.2$s,
              A = param.2$A,
```

```
                W = param.2$W,
                d = param.2$d,
                Q = param.2$Q,
                w = param.2$w,
                M = param.2$M,
                hi = param.2$hi,
                add = param.2$add,
                b = param.2$b,
                SIMPLIFY = F)

# Sample file as data.frame
s.2 <- do.call(rbind, s.1)

# frame.PSU
frame.PSU[, "cum.size"] <- cumsum(frame.PSU[, "size"])
frame.PSU[, "a"] <- frame.PSU[, "cum.size"] - frame.PSU[, "size"]
frame.PSU[, "b"] <- frame.PSU[, "cum.size"]


### Selecting of PSUs

# List of sampling points
s.3 <- s.2$A


### Function to select PSUs by sampling points
f.sel <- function(a, b, x) {
  a <- unlist(a)
  b <- unlist(b)
  x <- as.numeric(x[1])
  n <- length(a)
  return((1:n)[x > a & x < b])
}

# Indexes of sampled PSUs
s.4 <- unlist(sapply(s.3, f.sel, a = frame.PSU$a, b = frame.PSU$b))

if (length(s.4) != nrow(s.2))
  stop("Kļūda s.4 vai s.2 -- nav vienāds garums")

# PSU sample
s.5 <- data.frame(frame.PSU[s.4, ], s.2)

if (length(unique(s.5[, name.PSU])) < nrow(s.5))
  stop("Dublicate in PSU sample")

# PSU sample (IDs only)
s.6 <- s.5[, name.PSU]

param$total.m <- param$A * param$B * param$w

frame.PSU$sampled <- as.numeric(frame.PSU[, name.PSU] %in% s.6)

frame.PSU <- merge(frame.PSU, param[c("s", "m", "M", "total.m")],
                   by.x = name.strata, by.y = "s")
frame.PSU$m <- frame.PSU$m * frame.PSU$sampled

frame.PSU[, name.weight.s1] <- frame.PSU$M /
  (frame.PSU$size * frame.PSU$total.m) * frame.PSU$sampled


### Second-stage sampling

sample.TSU <- SamplingClusterStr(frame.TSU,
```

```
                                        frame.SSU,
                                        frame.PSU$m,
                                        name.weight.s2,
                                        name.SSU,
                                        name.PSU)

  sample.TSU <- merge(sample.TSU, frame.PSU[, c(name.PSU, name.weight.s1)])

  sample.TSU[, name.weight] <- sample.TSU[, name.weight.s1] *
    sample.TSU[, name.weight.s2]


  ### Add week

  s.week <- s.5[, c(name.PSU, "week")]
  sample.TSU <- merge(sample.TSU, s.week)
  names(sample.TSU)[ncol(sample.TSU)] <- name.week

  return(sample.TSU)
}
```

1.2 Calculation of Interviewing Expenses

1.2.1 TSP Solver for a Single Interviewer

The name of procedure is `Trip`. The aims of the procedure are:

- Solve a TSP for an interviewer.
- Calculate the distance of a trip for an interviewer.

The arguments of the procedure:

- `data`: two-column matrix $\boldsymbol{A}$ where each row of the matrix represents the coordinates of sampled dwellings. The number of rows is denoted by n.
- `coord.int`: vector $\boldsymbol{v}$ (length two) with the coordinates of the residences of interviewer.

The steps of the procedure:

- The package `TSP` is loaded (Hahsler & Hornik, 2011).
- Converting class of the arguments if necessary.
- Vector $\boldsymbol{v}$ is added to the matrix $\boldsymbol{A}$ as the first row of the matrix.
- The lower triangle of a distance matrix is computed by the R function `dist` (R Core Team, 2013) from the coordinates defined by $\boldsymbol{v}$ and $\boldsymbol{A}$. Euclidean distance is used.
- Symmetric TSP is created by the R function `TSP` from the package `TSP` (Hahsler & Hornik, 2011) using the lower triangle of a distance matrix.
- TSP is solved by the R function `solve_TSP` from the package `TSP` (Hahsler & Hornik, 2011) using the nearest insertion algorithm. The residence of interviewer is set as the first node when solving the TSP.
- The output of the procedure is a list with two objects:
  - The distance of the trip as the value of the attribute `tour_length` extracted from the solution of the TSP.

– The coordinates of the trip sorted according to the solution of the TSP.

See Section 4.1 for more details. See Figure 1.1, Figure 1.2 and Figure 1.3 for the examples of the results.

```
Trip <- function(data, coord.int) {

  # Libs
  require(TSP)

  # Argument type convertion
  data <- as.data.frame(data[1:2])
  coord.int <- as.numeric(coord.int)[1:2]

  data <- rbind(coord.int, data)

  dist.matrix <- dist(data)
  tsp <- TSP(dist.matrix)
  t <- solve_TSP(tsp, control = list(start = 1L))

  data <- as.data.frame(rbind(data[t, ], coord.int))

  rownames(data) <- NULL

  return(list(attr(t, "tour_length"), data))
}
```

There is another procedure `Trip.fast`. The procedure `Trip.fast` is equal to the procedure `Trip`. The difference is the result of the `Trip.fast` – only the distance of the trip is returned.

```
Trip.fast <- function(data, coord.int) {

  # Libs
  require(TSP)

  # Argument type convertion
  data <- as.data.frame(data[1:2])
  coord.int <- as.numeric(coord.int)[1:2]

  data <- rbind(coord.int, data)

  dist.matrix <- dist(data)
  tsp <- TSP(dist.matrix)
  t <- solve_TSP(tsp, control = list(start = 1L))

  return(attr(t, "tour_length"))
}
```

1.2.2 TSP Solver for a Multiple Interviewers and Weeks

The name of procedure is `vTrip.fast.1`. The aims of the procedure are:

- Solve a TSP for multiple interviewers and weeks.
- Calculate the total distance of trips for multiple interviewers and weeks.

The arguments of the procedure:

- `data`: four-column matrix where each row of the matrix represents the coordinates of sampled dwellings. The columns are:

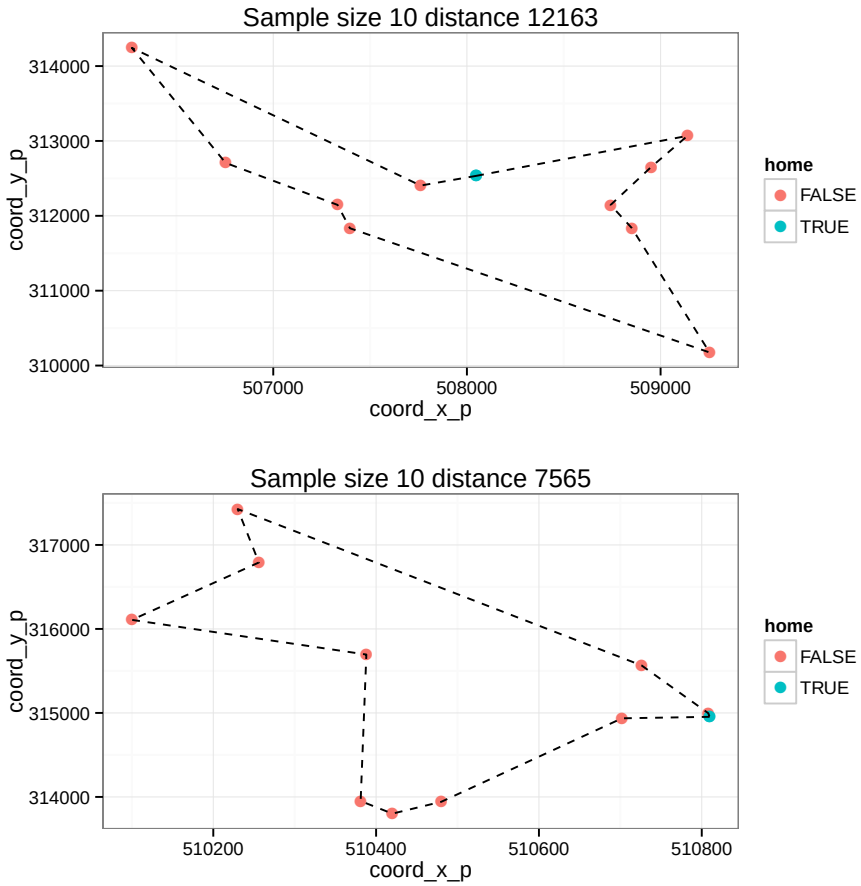


Figure 1.1 **Example 1 of the result of the procedure `Trip`**

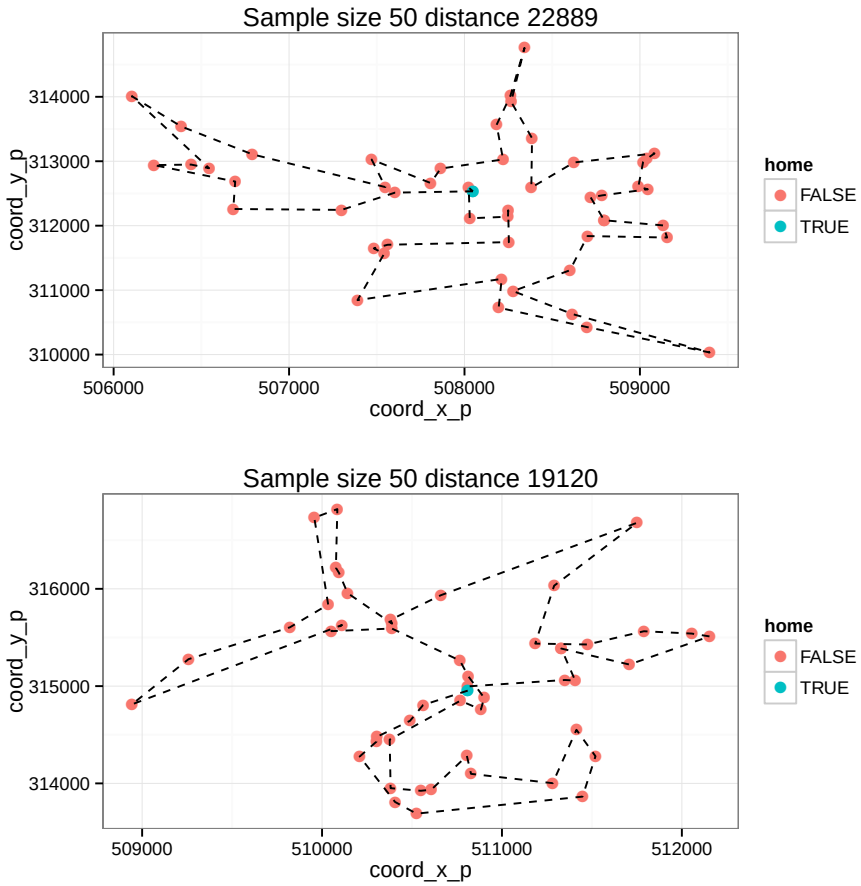


Figure 1.2 **Example 2 of the result of the procedure `Trip`**

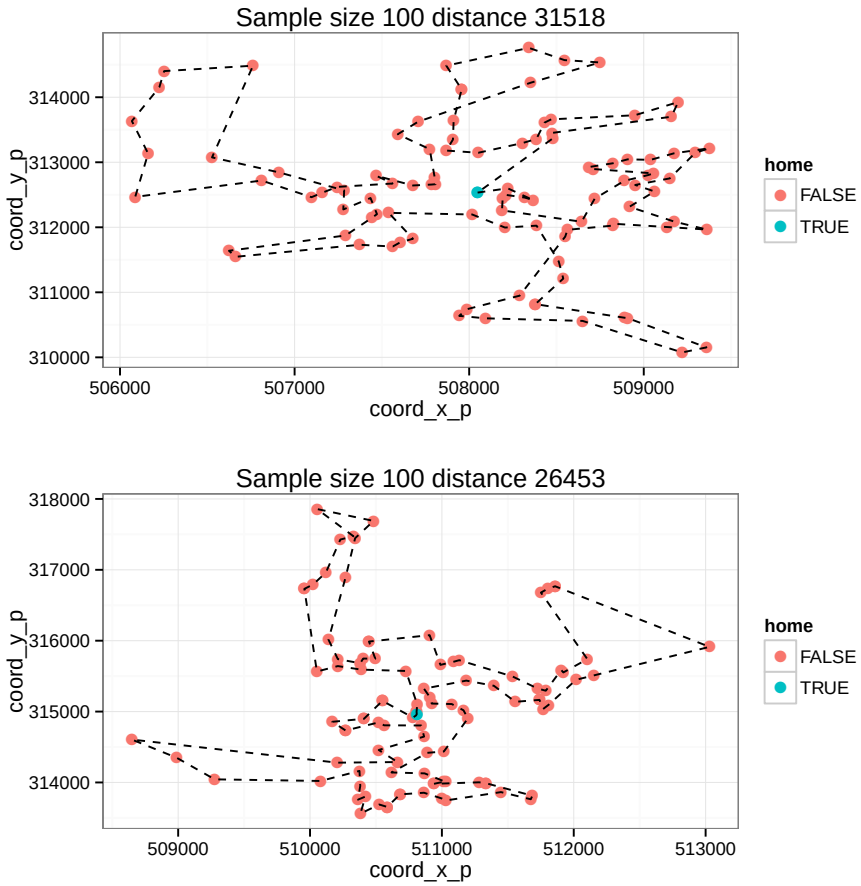


Figure 1.3 **Example 3 of the result of the procedure `Trip`**

  - The first column contain the labels of interviewers.
  - The second column contain the week numbers.
  - The third and fourth column contains the coordinates of the sampled dwellings.
- `coord.int`: three-column matrix where each row of the matrix represents the coordinates of interviewer residence. The columns are:
  - The first column contain the labels of interviewers.
  - The second and third column contains the coordinates of the interviewer residence.

The steps of the procedure:

- Converting class of the arguments if necessary.
- The distance of the trip is computed for each combination of an interviewer and a week available in the `data`. The distance is computed by the procedure `Trip.fast`.
- The sum of all distances is returned as the result of the procedure.

```
vTrip.fast.1 <- function(data, coord.int) {

  # Argument type convertion
  data <- as.data.frame(data[1:4])
  coord.int <- as.data.frame(coord.int)[1:3]

  # Redefine Trip
  Trip.alt <- function(int, week) {
    r <- Trip.fast(data[data[,1] == int & data[,2] == week, 3:4],
                   coord.int[coord.int[,1] == int, 2:3])
    return(r)
  }

  # Run
  tab <- unique(data[1:2])
  res <- mapply(Trip.alt, tab[,1], tab[,2])

  return(sum(res))
}
```

1.2.3 Calculation of Interviewing Expenses

The name of procedure is `Cost`. The aim of the procedure is:

- Calculate the field work expenses.

The arguments of the procedure:

- `trip`: the total distance done by interviewers during the field work operation denoted by d (in kilometres).
- `cons`: average fuel consumption during the field work operation denoted by K_f (in litres per kilometre).
- `price.f`: average fuel price during the field work operation denoted by C_f (in lats per litre).
- `k.d`: user specified distance adjustment coefficient (default value is 1).
- `n.h`: the total number of households interviewed denoted by m.
- `n.p`: the total number of individuals interviewed denoted by n.

- `price.h`: the cost per household questionnaire denoted by C_h.
- `price.p`: the cost per individual questionnaire denoted by C_p.

All arguments of the procedure can be defined as scalars or vectors with a length n. The result of the procedure will be scalar, if all arguments are scalars. The result will be a vector with length n, if at least one argument is a vector with length n. The steps of the procedure:

- Converting class of the arguments if necessary.
- The result of the procedure is computed as $d \cdot K_f \cdot C_f \cdot K_d + m \cdot C_h + n \cdot C_p$ if all arguments are scalars.
- The result of the procedure is computed as $\boldsymbol{d} \odot \boldsymbol{K}_f \odot \boldsymbol{C}_f \odot \boldsymbol{K}_d + \boldsymbol{m} \odot \boldsymbol{C}_h + \boldsymbol{n} \odot \boldsymbol{C}_p$ if at least one argument is vector. All scalar arguments are converted to vectors (size n) with all elements equal to the value of scalar argument.

```
Cost <- function(trip = 0,
                 cons = 0,
                 price.f = 0,
                 k.d = 1,
                 n.h = 0,
                 n.p = 0,
                 price.h = 0,
                 price.p = 0) {

  trip <- as.numeric(trip)
  cons <- as.numeric(cons)
  price.f <- as.numeric(price.f)
  k.d <- as.numeric(k.d)
  n.h <- as.numeric(n.h)
  n.p <- as.numeric(n.p)
  price.h <- as.numeric(price.h)
  price.p <- as.numeric(price.p)

  return(trip * cons * price.f * k.d + n.h * price.h + n.p * price.p)
}
```

1.3 Estimation of Population Parameters

1.3.1 Estimation of The Primary Population Parameters

Primary population parameters are parameters estimated directly from individual data. The name of the procedure is `Estimation`. The aim of the procedure is:

- Estimate the primary population parameters from sample data.

The arguments of the procedure:

- `x`: Matrix of individual data where rows represent sampling units and columns represent variables. The number of rows of `x` is denoted by n.
- `w`: Vector with length 1 or n of weights denoted by $\boldsymbol{w}$. If length is 1, constant weight is assumed.
- `param`: The matrix with three columns. Each row represent a parameter to be estimated. The number of rows is denoted by m. The columns are:

  - The first column contains the type of estimator: `total`, `mean` or `ratio`.
  - The second column contains the name of variable used in total or mean estimator or the name of variable denoted by $\boldsymbol{y}$ used in numerator of ratio estimator.
  - The third column contains a value `NA` if total or mean estimator is used or the name of variable denoted by z used in denominator of ratio estimator.

The steps of the procedure:

- The package `bigmemory` is loaded. It is necessary to manipulate arguments with the `big.matrix` class (Kane & Emerson, 2012a).
- Converting class of the arguments if necessary.
- Testing the coherence of the arguments.
- Sample size n is computed as the number of rows of the `x`.
- Population size N is estimated as

$$\hat{N} = \begin{cases} nw & \text{if } |\boldsymbol{w}| = 1, \\ \sum_{i=1}^{n} w_i & \text{if } |\boldsymbol{w}| > 1. \end{cases}$$

- Vector $\boldsymbol{E}$ is created with length $m + 2$.
- $E_1 = \hat{N}$
- $E_2 = n$
- $E_k, k > 2$ is computed for each row of `param`:

$$E_k = \begin{cases} \sum_{i=1}^{n} w_i y_i & \text{if parameter is total,} \\ \frac{\sum_{i=1}^{n} w_i y_i}{\hat{N}} & \text{if parameter is mean,} \\ \frac{\sum_{i=1}^{n} w_i y_i}{\sum_{i=1}^{n} w_i z_i} & \text{if parameter is ratio.} \end{cases}$$

- Vector $\boldsymbol{E}$ is returned as the result of the procedure.

```
Estimation <- function(x, w, param) {

  # Libs
  require(bigmemory)

  # Argument type convertion
  if (!is.big.matrix(x)) x <- as.data.frame(x)
  w <- as.vector(as.numeric(w))
  param <- as.matrix(param)

  # Testing
  if (length(w) != 1 & length(w) != nrow(x)) stop("Error in w")
  if (!all(param[,1] %in% c("sum", "mean", "ratio"))) stop("Error in param 1")
  if (ncol(param) != 3) stop("Error in param 2")

  # Estim
  n <- nrow(x)
  N <- ifelse(length(w) == 1, w * n, sum(w))

  E <- as.data.frame(matrix(NA, 1, 2+nrow(param)))

  E[1,1] <- N
```

```
  E[1,2] <- n
  colnames(E)[1:2] <- c("N", "n")

  for (i in 1:nrow(param)) {
    a <- sum(w * x[,param[i,2]])
    if (param[i,1] == "sum") estim <- as.matrix(a)
    if (param[i,1] == "mean") estim <- as.matrix(a / N)
    if (param[i,1] %in% c("sum","mean"))
      nam <- paste(param[i,1], ".", param[i,2], sep="")
    if (param[i,1] == "ratio") {
      b <- sum(w * x[,param[i,3]])
      estim <- as.matrix(a / b)
      nam <- paste("r", param[i,2], param[i,3], sep=".")
      }
    E[1,2+i] <- estim
    colnames(E)[2+i] <- nam
    }

  return(E)
}
```

1.3.2 Estimation of The Secondary Population Parameters

The secondary population parameters are the parameters computed from the primary population parameters. This procedure is specific procedure for employment parameters used in the work. The name of the procedure is `CompEmp`. The aim of the procedure is:

- To compute the estimates of five specific secondary employment parameters using three primary population parameters.

The primary population parameters used in computation of the secondary population parameters:

- Number of employed individuals.
- Number of unemployed individuals.
- Number of inactive individuals.

The secondary population parameters computed:

- Number of individuals in working age (15–74).
- Number of active individuals.
- Activity rate.
- Employment rate.
- Unemployment rate.

The arguments of the procedure:

- `x`: Matrix containing three columns with the estimates of the primary population parameters.
- `var.names`: Vector with length 3 with the names of the variables in `x` containing the estimates of the primary population parameters. The order of variables names is important. The order should match the order of variables mentioned before.

The steps of the procedure:

- Converting class of the arguments if necessary.

- Testing the coherence of the arguments.
- Five estimates of the secondary population parameters are computed and attached to `x`.
- `x` is returned as the result of the procedure.

```
CompEmp <- function(x, var.names) {

  x <- as.data.frame(x)
  if (is.null(dim(x))) stop("x has to be two-dimensional")

  var.names <- as.character(var.names)[1:3]

  x$sum.pop <- rowSums(x[var.names])
  x$sum.act <- rowSums(x[var.names[1:2]])

  x$r.act <- x$sum.act / x$sum.pop
  x$r.empl <- x[, var.names[1]] / x$sum.pop
  x$r.unempl <- x[, var.names[2]] / x$sum.act

  return(x)
}
```

1.4 Other Functions

1.4.1 Extraction of Data From The Dynamic Population Data

The name of the procedure is `extr.data`. The aim of the procedure is:

- To extract the data from the dynamic population data according to the sampled units and weeks.

The arguments of the procedure:

- `x`: the data frame of dynamic population where rows are sampling units and columns are weeks (see Section 2.2 for more details).
- `rows`: vector of sampled units defined by the row number.
- `cols`: vector of week numbers for sampled units.
- `col.skip`: number of columns to be skipped from the dynamic population.
- `var.name`: variable name to be used for the resulting variable.

The steps of the procedure:

- Converting class of the arguments if necessary.
- Testing the coherence of the arguments.
- Necessary data are extracted from the `x` and saved as a variable `d2`.
- `d2` is merged with the first columns of `x` if `col.skip` > 0 and saved as data frame `d`.
- `d` is returned as the result of the procedure.

```
extr.data <- function(x, rows, cols, col.skip = 0, var.name = "var") {

  rows <- as.integer(rows)
  cols <- as.integer(cols)
  col.skip <- as.integer(col.skip)[1]
  var.name <- as.character(var.name)[1]
```

```
7
8     N <- nrow(x)
9     M <- ncol(x) - col.skip
10
11    if (is.null(dim(x))) stop("x is not a two-dimensional object")
12
13    if (min(rows) < 1) stop("wrong id for rows")
14    if (max(rows) > N) stop("wrong id for rows")
15    if (min(cols) < 1) stop("wrong id for cols")
16    if (max(cols) > M) stop("wrong id for cols")
17
18    if (length(rows) != length(cols))
19      stop("rows and cols has to be the same length")
20
21    extr <- function(r, c) x[r, c]
22    d2 <- mapply(extr, rows, cols + col.skip)
23
24    if (col.skip > 0) {
25      d <- data.frame(data.frame(x[rows, 1:col.skip]), d2)
26    } else {
27      d <- data.frame(d2)
28    }
29
30    colnames(d)[col.skip + 1] <- var.name
31
32    return(d)
33  }
```

1.4.2 Monte Carlo Simulations

Parallel computing is used to run Monte Carlo simulation. Parallel computing allows to reduce execution time of the simulation because each iteration is independent and can be run in parallel. The name of the procedure is `Sim`. The aim of the procedure is:

- To run Monte Carlo simulation and save the results of each iteration.

The arguments of the procedure:

- `fun`: function to be run in a simulation.
- `arg`: the matrix of arguments to be passed to the function `fun` where rows are different sets of arguments and columns are the arguments of the `fun`
- `I`: number of iterations for each set of parameters (rows of `arg`).
- `name`: name for the resulting object and file where the results are saved.
- `print`: logical. Defined if the results of the simulation are printed.
- `log`: logical. Defined if log file is created.
- `seed`: can be used to define seed for the random generator. If `seed` is `NA` random seed is used.
- `cores`: the number of CPU cores to be used in the simulation.

The steps of the procedure:

- The R packages necessary for parallel computing `foreach` (Revolution Analytics, 2012b) and `doMC` (Revolution Analytics, 2012a) are loaded.

- Multicore parallel backend is registered by the function `registerDoMC` (Revolution Analytics, 2012a) with the number of CPU cores to be used defined by `cores`.
- Converting class of the arguments if necessary.
- Testing the coherence of the arguments.
- A test run is done by running the function `fun` with the first set of arguments from `arg`.
- Simulations are run in nested loop where the first loop goes from `1` to the number of rows in `arg` and the second loop goes from `1` to `I`.
- The function `fun` with corresponding arguments from `arg` are executed each iteration. `fun` is executed with `try` wrapper (R Core Team, 2013). If the execution of `fun` was error free, the results are saved in a data.frame. If the error occurred during the execution of `fun`, the error message is saved. `try` is used because it allows to run full simulation even some iterations result with an error.
- Execution time of simulation is computed.
- Results of simulation are saved in the data file in the working directory with the name `name` and extension `.Rdata`.
- The result of the procedure is a list with two objects – data.frame with results of the simulation and execution time.

```
Sim <- function(fun, arg, I = 5,
                name = "res",
                print = F,
                log = F,
                seed = NA,
                cores = multicore:::detectCores()) {

  require(foreach)
  require(doMC)

  registerDoMC(cores = cores)

  # Argument type convertion

  fun <- as.character(fun)[1]
  arg <- as.list(arg)
  I <- as.integer(I)[1]
  name <- as.character(name)[1]
  print <- as.logical(print)[1]
  log <- as.logical(log)[1]

  # Testing
  if (I<=0) stop("I has to be 1 or larger")
  if (!is.na(seed) & length(seed) != I) stop("Wrong length of seed")

  # Function to convert arg as one-row data.frame
  transf <- function(x) data.frame(lapply(x, function(x) t(data.frame(x))))

  # Test run
  test <- do.call(fun, arg[[1]])
  m1 <- length(test)

  arg2 <- transf(arg[[1]])
  m2 <- ncol(arg2)

  cat("Simulation name:", name, "\n")
```

```
  cat("Number of iterations:", I, "\n")
  cat("Number of cores:", cores, "\n")

  filename <- name

  t1 <- Sys.time()

  R <- foreach(a = 1L:length(arg), .combine = rbind, .inorder = F) %:%
    foreach(i = 1L:I, .combine = rbind, .inorder = F) %dopar% {
      if (log) cat(as.character(Sys.time()), ", ",
                   i, " of ", I, ", ", round(i/I*100, 1), "%\n",
                   file = paste(filename, ".log", sep = ""),
                   sep = "", append = T)

      if (!is.na(seed)) set.seed(seed[i])

      tr <- try(do.call(fun, arg[[a]]), T)

      res <- data.frame(t1, name, a, i, seed[i], transf(arg[[a]]))
      rownames(res) <- NULL

      if (class(tr) == "try-error")
        res <- data.frame(res, tr[[1]], matrix(NA, 1, m1)) else
        res <- data.frame(res, NA, tr)

      colnames(res) <- paste("v", 1:(5 + m2 + 1 + m1), sep = "")

      res
    }

  t2 <- Sys.time()
  time.run <- as.numeric(t2 - t1, units="secs")

  colnames(R) <- make.names(c("timestamp", "name", "a", "i", "seed",
                   colnames(arg2), "err", colnames(test)), unique = T)
  rownames(R) <- NULL

  assign(name, R)
  save(list = name, file = paste(filename, ".Rdata", sep=""))

  cat("Total time:", time.run, "sec\n")
  cat("Average time:", time.run / I, "sec\n")

  if (print) {
    cat("Rows in results:", nrow(R), "rows", "\n")
    cat("First results:", "\n")
    print(head(R))
  }

  return(list(R, time.run))
}
```