

WORKING PAPERS SES

Clustering the Swiss Pension Register

Loyal Christine Lettry

Under the supervision of
Prof. Dr. L. Donzé

**N. 529
II. 2023**

Clustering the Swiss Pension Register

Under the supervision of Prof. Dr. L. Donzé

Working Paper

by

Layal Christine Lettry

Fribourg, February 27, 2023

All models are wrong, but some are useful.

George Box, 1976



Contents

Contents	5
List of Tables	7
List of Figures	9
List of Scripts	11
Acknowledgements	13
Abstract	15
Introduction	17
1 Variables Definition and Statistical Highlights	19
1.1 Variables Definition	19
1.2 Statistical Highlights	25
2 Methodology and Application	29
2.1 Methodology	30
2.1.1 Literature review on mixed-type data	31
2.1.2 Kamila Clustering Algorithm	33
2.1.3 Kamila Clustering Algorithm Formalisation	36
2.1.4 Prediction strength algorithm	36
2.2 Application in R	38
2.2.1 Preparation of the data	39
2.2.2 Determination of k^*	40
2.2.3 Splitting the Pension Register into k^* clusters	41
3 Results	43
3.1 Kamila estimated parameters	43
3.1.1 Best number of cluster k^*	43
3.1.2 Characteristics of the clusters	46
3.2 Distribution of the AADR among the clusters	48
Conclusion	55

Annexes	57
A.1 OASI monthly pension amount in function to AADR (pension system 2021) . . .	58
A.2 Descriptive Statistics	58
A.2.1 Cluster 1	58
A.2.2 Cluster 2	60
A.2.3 Cluster 3	61
A.2.4 Cluster 4	63
A.2.5 Cluster 5	64
A.3 Histograms	66
A.3.1 Female old-age pensioners	66
A.3.2 Other types of OASI female pensioners	68
A.3.3 Male old-age pensioners	70
A.3.4 Other types of OASI male pensioners	72
A.4 Clusters Contingency Table	74
A.5 Clusters Scatterplots of the AADR	80
A.6 Clusters Scatterplots of the monthly pension amount	86
A.7 Clusters Summary Statistics	92
Scripts	97
Parameters Files	123
Acronyms	127
Acronyms	127
Bibliography	129
Printed References	129

List of Tables

1.1	Variables definition	24
1.2	Summary Statistics of the Pension Register	27
1.3	Summary Statistics of the Pension Register's Variables Recoded for NAs	27
2.1	Definition of the used categorical variables	40
2.2	Definition of the used continuous variables	41
3.1	Log-likelihood and Prediction Strength analysis of explanatory variables sets	44
3.2	Clusters Characteristics	46
3.3	Clusters Distribution	47
3.4	Table of Moments pro Cluster	49
A.1	Clusters Contingency Table	75
A.2	Summary Statistics of the Clustered Pension Register, Cluster 1	92
A.3	Summary Statistics of the Clustered Pension Register, Cluster 2	93
A.4	Summary Statistics of the Clustered Pension Register, Cluster 3	94
A.5	Summary Statistics of the Clustered Pension Register, Cluster 4	95
A.6	Summary Statistics of the Clustered Pension Register, Cluster 5	96
C.1	rrclust parameters file PARAM_KAMILA	125
C.2	rrclust parameters file PARAM_GLOBAL	126

List of Figures

1.1	OASI monthly pension amount in function to AADR (pension system 2020)	21
2.1	Kamila Clustering Algorithm	36
2.2	Workflow of the package rrclust	39
3.1	Prediction Strength Values for Determining the Optimal Number of Clusters k^*	45
3.2	Histograms of the AADR and of the monthly pension amount	50
3.2a	Frequency of the natural logarithm of AADR	50
3.2b	Frequency of the monthly pension amount	50
3.3	Density of the AADR and of the monthly pension amount	51
3.3a	Density of the natural logarithm of AADR	51
3.3b	Density of the monthly pension amount	51
3.4	Empirical Cumulative Distribution Function of the AADR and of the monthly pension amount	52
3.4a	ECDF of the natural logarithm of AADR	52
3.4b	ECDF of the monthly pension amount	52
3.5	Normal Quantile-Quantile plots of the AADR and of the monthly pension amount	53
3.5a	Normal Quantile-Quantile plot of the AADR	53
3.5b	Normal Quantile-Quantile plot of the monthly pension amount	53
A.6	OASI monthly pension amount in function to AADR for a minimal pension amount of 1195 CHF in effect in the year 2021	58
A.7	Histogram for divorced female old-age pensioners	66
A.8	Histogram for married female old-age pensioners	66
A.9	Histogram for single female old-age pensioners	67
A.10	Histogram for widowed female old-age pensioners	67
A.11	Histogram for divorced female pensioners getting another type of OASI pension	68
A.12	Histogram for married female pensioners getting another type of OASI pension	68
A.13	Histogram for single female pensioners getting another type of OASI pension	69
A.14	Histogram for widowed female pensioners getting another type of OASI pension	69
A.15	Histogram for divorced male old-age pensioners	70
A.16	Histogram for married male old-age pensioners	70
A.17	Histogram for single male old-age pensioners	71
A.18	Histogram for widowed male old-age pensioners	71
A.19	Histogram for divorced male pensioners getting another type of OASI pension	72
A.20	Histogram for married male pensioners getting another type of OASI pension	72

A.21 Histogram for single male pensioners getting another type of OASI pension . . .	73
A.22 Histogram for widowed male pensioners getting another type of OASI pension	73
A.23 Male AADR distribution according to the age	81
A.23a Males getting a type of pension different from old-age: AADR distribution	81
A.23b Male Old-age insurance AADR distribution	81
A.24 Female AADR distribution according to the age	82
A.24a Female getting a type of pension different from old-age: AADR distribution	82
A.24b Female Old-age insurance AADR distribution	82
A.25 Male AADR distribution according to the scale	83
A.25a Males getting a type of pension different from old-age: AADR distribution	83
A.25b Male Old-age insurance AADR distribution	83
A.26 Female AADR distribution according to the scale	84
A.26a Females getting a type of pension different from old-age: AADR distribution	84
A.26b Female Old-age insurance AADR distribution	84
A.27 Male and Female AADR distribution according to the age of retirement	85
A.27a Male Old-age insurance AADR distribution	85
A.27b Female Old-age insurance AADR distribution	85
A.28 Male monthly pension amount distribution according to the age	87
A.28a Males getting a type of pension different from old-age: monthly pension amount distribution	87
A.28b Male Old-age insurance monthly pension amount distribution	87
A.29 Female monthly pension amount distribution according to the age	88
A.29a Females getting a type of pension different from old-age: insurance monthly pension amount distribution	88
A.29b Female Old-age insurance monthly pension amount distribution	88
A.30 Male monthly pension amount distribution according to the scale	89
A.30a Males getting a type of pension different from old-age: insurance monthly pension amount distribution	89
A.30b Male Old-age insurance monthly pension amount distribution	89
A.31 Female monthly pension amount distribution according to the scale	90
A.31a Females getting a type of pension different from old-age: monthly pension amount distribution	90
A.31b Female Old-age insurance monthly pension amount distribution	90
A.32 Male and Female monthly pension amount distribution according to the age of retirement	91
A.32a Male Old-age insurance monthly pension amount distribution	91
A.32b Female Old-age insurance monthly pension amount distribution	91

List of Scripts

B.1	Wrapper for the data preparation and the Kamila algorithm execution	99
B.2	Wrapper for the preparation of the data	100
B.3	Preparation of the RR_OASI tibble	102
B.4	Training set and validation set	106
B.5	Splitting the variables according to their types	108
B.6	Wrapper for the Kamila algorithm execution part	112
B.7	Finding the parameter kstar	115
B.8	Splitting the register of rents	119
B.9	Log of the executed run	122

Acknowledgements

First of all, I would like to thank Prof. Dr. L. Donzé for his supervision, support and valuable advice. Then, I am very grateful to the **FSIO** for allowing me to use the Swiss **Pension Register (CCO/FSIO)** and to my former colleagues for their support during the five years I worked at the **FSIO** as an econometrician. I am pleased to have been able to use the **KAMILA clustering method** and the respective R package for the purpose of this paper and I want to thank the authors. Last but not least, I would like to thank my husband for his support and for proofreading this working paper.



Abstract

The anonymous data of the Swiss **Pension Register (CCO/FSIO) (PR)** are typically used to estimate (in the short, middle and long term) the future revenues and expenditures of the **Old-Age and Survivors' Insurance (OASI)**. In this perspective, it is essential to have a clear look at the register's main statistical features. To better understand it and benefit more from its richness, we propose analysing the raw data by an appropriate clustering method.

We face three main difficulties:

1. As not only continuous but also nominal or categorical variables structure the register, we have to choose a clustering method that considers any types of variables;
2. The a priori number of clusters should be in the first step determined, and thus the question of how to fix it is essential;
3. The method should run over big data.

Recently, A. Foss et al. (2016) and A. H. Foss and Markatou (2018a) proposed the **KAMILA clustering method** (KAY-means for MIXed LARge data), which is specifically designed to manage a clustering process for mixed distributions. Furthermore, a simple rewriting of the **KAMILA** algorithm allows an easy implementation in a map-reduce framework like Hadoop, thus being run on very large data sets. On the other hand, Tibshirani and Walther (2005) advocate the use of the "Prediction Strength" as a measure to find the optimal number of clusters.

We applied the **KAMILA clustering method** on the more than 2 000 000 observations of the **PR** data. Due to memory limit problems and time restrictions in our research, we did not use the Hadoop framework to run the **KAMILA** algorithm and had to work on a reduced sample size (but still very large and therefore sufficient to obtain satisfactory results). We were able to determine the optimal number of clusters, namely k^* , thanks to the **KAMILA** technique, while maintaining a reasonable computing time thanks to the smaller sample. We then applied the **KAMILA** algorithm on the whole Swiss **Pension Register (CCO/FSIO) (PR)** according to the optimal number of clusters k^* retrieved from the smaller sample.

On this basis, we analysed the partition of our data. The principal features of each cluster were described. As a result, we highlighted, in particular in the table 3.3 which summarises the results of the contingency table A.1, the similarities and dissimilarities between the **OASI** pensioners subgroups according to their sociodemographic characteristics. These pieces of information are helpful to better understand the structure of the Swiss **Pension Register (CCO/FSIO)**.



Introduction

In a first step, we describe the data of the Swiss Pension Register ¹ (CCO/FSIO ²) 2020 from a statistical point of view. In a second step, we present the clustering method we use to split the Swiss PR (CCO/FSIO), namely the KAMILA clustering method. Finally, we explain the implementation of this method on our data and we analyse the results.

The anonymous data of the Swiss PR (CCO/FSIO) are typically used to estimate (in a short, middle and long term) the future revenues and expenditures of the OASI. In order to better understand this register and to benefit more from its richness, we analyse the raw data as well as the clustered data.

Due to the large size of the Swiss PR (CCO/FSIO) and to the heterogeneity of its variables, we have to choose a clustering method adapted for mixed-type data allowing to handle a very large number of observations. We find that the KAMILA clustering method is the most suited clustering method for our data and we explain theoretically and empirically how it enables us to obtain the best number of clusters (denoted by k^*) for our dataset.

In chapter 1, the original dataset and its data are described by means of summary statistics. We then present theoretically the KAMILA clustering method as well as the way we implement it in the R software in chapter 2. In chapter 3, the resulting clusters are described statistically and graphically.

¹We denote Pension Register with its acronym PR further.

²Please consult the website: www.bsv.admin.ch.



Chapter 1

Variables Definition and Statistical Highlights

The Swiss PR (CCO/FSIO) contains the sociodemographic and economic characteristics and specific data of the Old-Age and Survivors' Insurance and of the Disability Insurance ¹. This register comes from the Central Compensation Office (CCO) ² located in Geneva (Switzerland). It contains all the data of the OASI and of the DI. The CCO makes it accessible for the employees of the the Federal Social Insurance Office (FSIO) via their internal server.

The variables of interest are Annual Average Determinant Revenue (AADR) and the monthly pension amount received by the insured persons. The first OASI pension amount is determined by using the AADR as an input in the equation (1.1), where the AADR reflects the whole life of an individual from an economic point of view. For this study, we focus on the observed year 2020.

We define a selection of variables from the Swiss PR (CCO/FSIO) in the section 1.1 and describe them in the section 1.2 by means of summary statistics.

1.1 Variables Definition

The table 1.1 defines the variables selected out of the Swiss Pension Register (CCO/FSIO). In this table, one can find the variable `year` which indicates the chosen year of the Swiss Pension Register (CCO/FSIO) which we use for our study. In our case, the variable `year` takes the value 2020. Our Swiss Pension Register sample reflects then the state of the Swiss retirement in the year 2020.

¹We will denote Old-Age and Survivors' Insurance and Disability Insurance with their respective acronyms OASI and DI further.

²Please consult the website www.zas.admin.ch.



The variables `age` and `age_retire` inform respectively on the age of an individual and on the age at which this person actually retired. The other sociodemographic variables such as nationality (`nat`), residence (`resid`) and sex (`sex`) complete the information needed to characterise each individual.

There are two categorical (nominal) variables, i.e. `marital_stat` and `benef_type` defining respectively the marital status and the pension type of an individual, which were converted into dummies for each category. All these dummy variables are defined in the same table and were created in order to build a design matrix for estimating the parameter k^* (cf. chapter 3). Note that the pension type, indicated by `benef_type`, is not only due to the old-age insurance but also to survivor insurance like in the case of spouse's loss (i.e. widow pension), of mother's or father's loss (i.e. mother's or father's orphan pension) or of both mother's and father's loss (i.e. twice orphan pension). In addition to these cases, there existed, until the 10th revision of the **OASI** (1997), a spouse's complementary pension for the younger wife of a retired man who could ask for it because she was not yet allowed to benefit from a old-age pension due to her age. This pension type still appears in the **Pension Register (CCO/FSIO)** because, once it has been granted to an individual (before 1997), it cannot be suppressed³. In case of a retired man or woman with a child less than 18 years old or between 18 and 25 years old and still studying, a father's child pension or a mother's child pension can be granted. The **DI** pension is also a part of the **Pension Register (CCO/FSIO)** but, as we concentrate only on the **OASI**, we exclude the data related to the **DI**.

The variable `aadr`, whose natural logarithm is given by `ln_aadr`, gives the **Annual Average Determinant Revenue (AADR)** used to determine the first pension amount of an individual leaving the labour market once he decided to retire. This variable is the one we are interested in because we want to be able to retrieve the most probable **AADR** by only relying on the sociodemographic and economic characteristics of the **OASI** new beneficiaries. The **AADR** determines the pension amount according to the functional relationship formalised by the equation (1.1) and depicted in the figure 1.1 for the minimal pension amount of 1185 CHF in effect in the year 2020 and in accordance with our dataset⁴.

By predicting the **AADR** for the projected number of **OASI** beneficiaries⁵, it will be possible to retrieve the pension amount they will receive once they get retired thanks to equation (1.1). Let mr be the minimal pension amount in the year of the occurrence of the insured event t , according to the **article 34, LAVS**, the **OASI** monthly pension amount for an individual i with a scale equal to 44 is defined as

$$\text{monthly_pension}_{t,i} = si \cdot \begin{cases} mr_t & \text{if } aadr_i \leq mr_t \cdot 12, \\ mr_t \cdot \frac{74}{100} + aadr_i \cdot \frac{13}{600} & \text{if } mr_t \cdot 12 < aadr_i \leq mr_t \cdot 36, \\ mr_t \cdot \frac{104}{100} + aadr_i \cdot \frac{8}{600} & \text{if } mr_t \cdot 36 < aadr_i < mr_t \cdot 72, \\ mr_t \cdot 2 & \text{if } aadr_i \geq mr_t \cdot 72. \end{cases} \quad (1.1)$$

³Please consult the **FSIO publication relative to the 10th revision of the OASI (1997)** (last consulted on the 03.11.2021).

⁴The minimal pension amount in effect from the 1st of January 2021 is equal to 1195 CHF (i.e. pension system 2021). The corresponding functional relationship is shown in the figure A.6.

⁵According to the population scenarios obtained from the **Swiss Statistical Federal Office**.

with si ⁶ being the factor for incomplete pensions depending on $scale_{e_i,i}$ — defined in the equation (1.2) — and where `monthly_pension` and `aadr` are defined in the table 1.1.

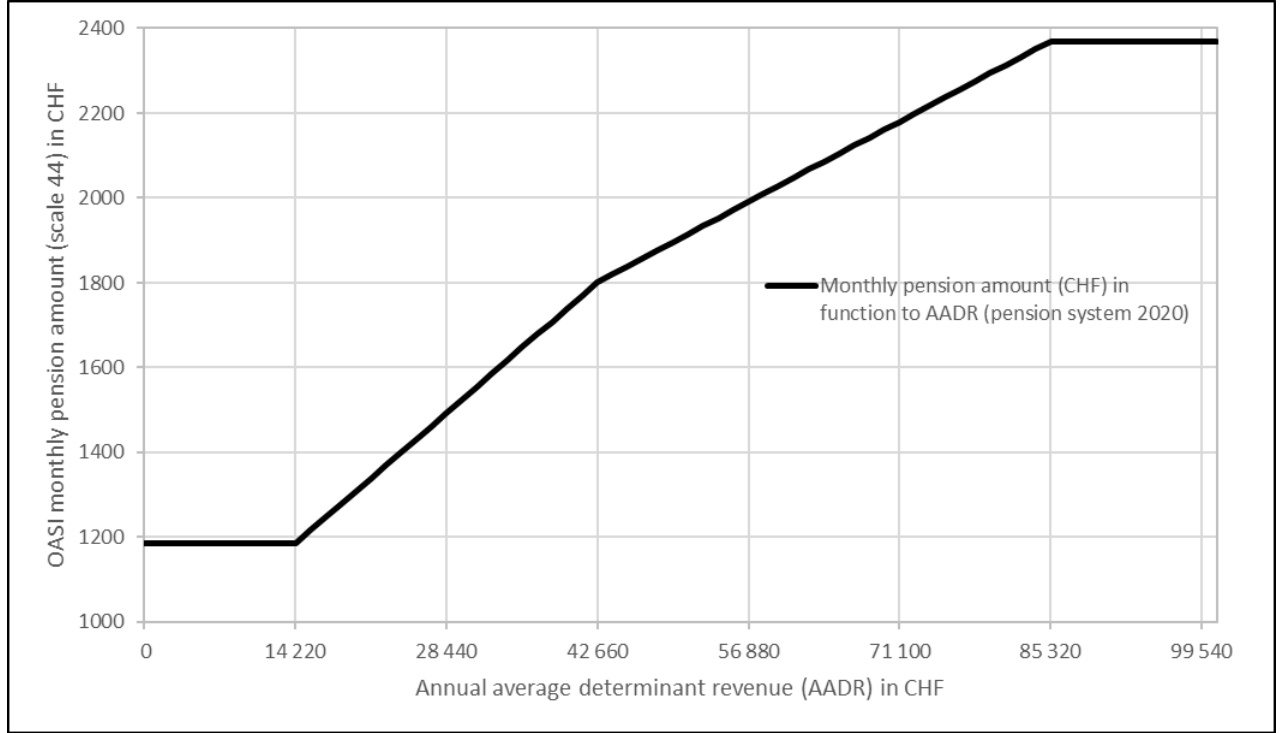


Figure 1.1: OASI monthly pension amount in function to AADR (pension system 2020)

As written in the table 1.1, the variable `scale` is defined as a factor relative to the number of contribution years to the OASI. This factor is determined by the maximal total number of OASI contribution years, to the individual's effective total number of OASI contribution years — given by the variable `contrib_m_ind` divided by 12 months — and to the number of OASI contribution years corresponding to the individual's age group — indicated by the variable `contrib_y_ageclass` defined in the equation (1.3). We can express this functional relationship with the equation (1.2):

$$scale_{e_i,i} = \frac{44 \cdot \frac{contrib_m_ind_{e_i,i}}{12}}{contrib_y_ageclass_i}, \quad (1.2)$$

where

$$contrib_y_ageclass_i = e_i - year_of_birth_i - 21 \quad (1.3)$$

⁶The factor si is equal to 1 if $scale_{e_i,i} = 44$, corresponding to a complete pension. Please consult the table 11 of the OASI and DI Pensions Tables (2019).

with e_i defining the year of the occurrence of the insured *event* of the individual i , `year_of_birth` the individual i 's year of birth and where 21 is the age at which **OASI** contributions must be paid for the first time (according to the pension system 2020).⁷

Note that the monthly pension amount given by the variable `monthly_pension` defined in the equation (1.1) will then be analysed in order to know if it must be capped or not. This is the case for the monthly pension amount of each of the couple members if sum of both pensions exceed 150% of the maximal pension amount — in 2020, the maximal pension amount is 2370 CHF. The old-age pension amount will be **capped** for each of member of the couple if the sum of their benefits exceeds 3555 CHF. In that case, the variable `capping` would be equal to 1.

The revenues earned by each of the couple members over the years under wedding contract will be **split** when both of them are getting retired or divorced and when an old-age pension is granted to a widow. In that case, the variable `splitting` is equal to 1.

At the splitting time or in the year of the occurrence of the insured event, **the bonus for childcare** and **the bonus for caregiving** are distributed. Their respective amounts are computed on the basis of the number of months dedicated to each of these activities, respectively given by the variables `bonus_m_edu` and `bonus_m_assist`. This fictive revenue is added to the total income of the individual which is used to determine his pension amount. This amount is equal (for both types of bonus) to three times the minimal old-age annual pension amount in effect in the year of the occurrence of the insured event.

⁷Please consult the tables 1 and 2 of the **OASI and DI Pensions Tables (2019)**.

Variable	Definition	Values
year	Year subset of the Pension Register (numeric)	2020;
nat	Nationality (dummy)	1 if foreign, 0 if Swiss;
resid	Residence (dummy)	1 if foreign country, 0 if Switzerland;
sex	Sex (dummy)	1 if female, 0 if male;
age	Age (numeric)	0, 1, ..., 98, 99;
age_retire	Observed retirement age (numeric)	62, 63, ..., 69, 70, NA for other pension types;
marital_stat	Marital status (categorical)	1 if divorced, 2 if single, 3 if married, 4 if widowed;
marital_stat1	Marital status for divorced (dummy)	1 if divorced, 0 otherwise;
marital_stat2	Marital status for single (dummy)	1 if single, 0 otherwise;
marital_stat3	Marital status for married (dummy)	1 if married, 0 otherwise;
marital_stat4	Marital status for widowed (dummy)	1 if widowed, 0 otherwise;
scale	Factor relative to the number of contribution years to the OASI (numeric)	1, 2, ..., 43, 44;
eprc	Cumulated full pension amount equivalent (numeric)	$\frac{1}{44}, \frac{2}{44}, \dots, \frac{43}{44}, \frac{44}{44}$;
monthly_pension	Monthly pension amount in CHF (numeric)	Cf. Table 1.2;
aadr	Annual Average Determinant Revenue (AADR) (numeric)	Cf. Table 1.2;
ln_aadr	Natural logarithm of aadr (numeric)	Cf. Table 1.2;
benef_type	Type of pension (categorical)	1 if old-age, 2 if widow, 3 if father's orphan, 4 if mother's orphan, 5 if twice orphan, 6 if spouse's complementary pension amount, 7 if father's child, 8 if mother's child;
benef_type1	Old-age pension (dummy)	1 if old-age pension, 0 otherwise;
benef_type2	Widow pension (dummy)	1 if widow pension, 0 otherwise;
benef_type3	Father's orphan pension (dummy)	1 if father's orphan pension, 0 otherwise;
benef_type4	Mother's orphan pension (dummy)	1 if mother's orphan pension, 0 otherwise;
benef_type5	Twice orphan pension (dummy)	1 if twice orphan pension, 0 otherwise;
benef_type6	Spouse's complementary pension (dummy)	1 if spouse's complementary pension, 0 otherwise;
benef_type7	Father's child pension (dummy)	1 if father's child pension, 0 otherwise;

Table 1.1 continued from previous page

Variable	Definition	Values
benef_type8	Mother's child pension (dummy)	1 if mother's child pension, 0 otherwise;
capping	Ceiling pension amount (dummy)	1 if the pension amount is capped, 0 otherwise;
contrib_m_ind	Contribution period used to compute the AADR (total number of months, numeric)	Cf. Table 1.2;
contrib_y_ageclass	Total number of contribution years of the age class (numeric)	Cf. Table 1.2;
splitting	Splitting income (dummy)	1 if splitted income, 0 otherwise;
bonus_m_edu	Total number of months dedicated to childcare	Cf. Table 1.2;
bonus_m_assist	Total number of months dedicated to caregiving	Cf. Table 1.2.

Table 1.1: Variables definition

1.2 Statistical Highlights

The table 1.2 shows us the summary statistics of the whole Pension Register for the year 2020. These summary statistics describe the data of the whole dataset.

However, before analysing the variables `contrib_m_ind`, `contrib_y_ageclass`, `splitting`, `bonus_m_edu` and `bonus_m_assist`, we should note that the NA values of these variables were replaced by 0 since there were problems with applying the KAMILA clustering method if there were only NA values in a cluster. This was the case for the pensioners receiving a pension type different from the old-age pension. We had to recode these NA values into 0 in order to take into account these variables for our tests presented in the table 3.1. Therefore, the descriptive statistics of the table 1.2 correspond to the already modified variables whereas the table 1.3 shows the descriptive statistics of the respective raw variables. These recoded variables do not enter into the final set of explanatory variables. Those are listed in the tables 2.1 and 2.2.

As one can see, the Pension Register for the year 2020 contains 2 688 607 observations. The individuals are between 0 and 99 years old. As provided in the Swiss law LAVS, women can retire between 62 and 69 years old and men between 63 and 70 years old. Therefore, the minimum and maximum values for the variable `age_ret` are resp. 62 and 70. For this variable, there are less observations, i.e. 2 438 759, because values can be retrieved only for the individuals concerned with retirement.

It is clearly shown that the number of women is higher than the number of men the OASI given that the Pension Register in 2020 is constituted by 56.5% of female beneficiaries. Besides, 63.2% of the individuals are Swiss and 65.1% live in Switzerland. Divorced individuals constitute 12.5% of the sample, single ones 8.7% and widowed ones 24%. Married beneficiaries constitute the most common marital status as this concerns 54.8% of the population sample. As expected, the old-age pension is the most common among the population since 90.7% of the individuals receive it. Among the other pension types, the widow pension type is the most prevailing as it touches 6.3% of the population, followed by the father's child pension with 1%.

The scale ranges from 0 to 44, where 0 concerns mostly the children who benefit from an orphan pension and for whom no scale can be computed, whereas a scale equal to 44 yields to a complete pension. The variable `eprc` is simply the variable `scale` divided by its maximal possible value (44). The value 0.734 means that, in average, 73.4% of a complete pension is due by the Swiss state pension provision.

In average, the pensioners have paid contributions during 360 months, respectively 30 years. The maximal value of `contrib_m_ind` is 540 which corresponds to a 45 years of regularly paid contributions to the OASI from the age of 21 until the legal retirement age. The variable `contrib_m_ind` will then be compared with the number of contribution years of the age class belonging to the respective individual — given by the variable `contrib_y_ageclass` — in order to determine the scale as it is formally written down in the equation (1.2).

The AADR has a large standard deviation. Therefore, we cannot interpret its mean correctly, as it is largely influenced by extreme values. Nonetheless, we can say that less than

25% will get the maximal old-age pension amount (i.e. $2 \cdot 1185 = 2370$), provided that the individuals in question have the maximal scale.

There are 30.4% of the pensions which are capped and 59.7% of the revenues earned by each of the couple members over the years under wedding contract which are split in case of a divorce or a retirement as well as when a widow gets an old-age pension. By omitting the observations with NA values, 60.9% of the individuals were affected by the splitting of their revenues.

The average number of months dedicated to childcare is about 69 months and 109 months by excluding the NA values (cf. table 1.3). The maximal number of the variable `bonus_m_edu` is very high (504 months) but can be due to having several children with an important age difference. On average, one spends less than one month for caregiving but, by excluding the NA values (cf. table 1.3), we see that about 36 months are dedicated to assistance care on average. The maximum of the variable `bonus_m_assist` is 264 months.

Variable	N	Mean	Std. Dev.	Min	Pctl. 25	Pctl. 75	Max
year	2688607	2020	0	2020	2020	2020	2020
age	2688607	73.481	11.679	0	68	80	99
age_ret	2438759	64.324	0.784	62	64	65	70
sex	2688607	0.565	0.496	0	0	1	1
nat	2688607	0.368	0.482	0	0	1	1
resid	2688607	0.349	0.477	0	0	1	1
eprc	2688607	0.734	0.37	0	0.386	1	1
aadr	2688607	60677.978	54923.606	0	39816	72522	17647020
monthly_pension	2688607	1395.412	773.527	7	664	1903	3678
capping	2688607	0.304	0.46	0	0	1	1
contrib_m_ind	2688607	360.535	191.287	0	172	516	540
contrib_y_ageclass	2688607	41.26	5.631	0	42	44	45
splitting	2688607	0.597	0.491	0	0	1	1
bonus_m_edu	2688607	69.297	65.411	0	0	120	504
bonus_m_assist	2688607	0.091	2.429	0	0	0	264
benef_type	2688607	1.191	0.817	1	1	1	8
marital_stat	2688607	2.904	0.904	1	3	3	4
scale	2688607	32.29	16.271	0	17	44	44
marital_stat1	2688607	0.125	0.331	0	0	0	1
marital_stat2	2688607	0.087	0.282	0	0	0	1
marital_stat3	2688607	0.548	0.498	0	0	1	1
marital_stat4	2688607	0.24	0.427	0	0	0	1
benef_type1	2688607	0.907	0.29	0	1	1	1
benef_type2	2688607	0.063	0.243	0	0	0	1
benef_type3	2688607	0.009	0.094	0	0	0	1
benef_type4	2688607	0.003	0.051	0	0	0	1
benef_type5	2688607	0	0.003	0	0	0	1
benef_type6	2688607	0.008	0.088	0	0	0	1
benef_type7	2688607	0.01	0.099	0	0	0	1
benef_type8	2688607	0.001	0.024	0	0	0	1

Table 1.2: Summary Statistics of the Pension Register

Variable	N	Mean	Std. Dev.	Min	Pctl. 25	Pctl. 75	Max
raw_contrib_m_ind	2688309	360.575	191.26	12	172	516	540
raw_contrib_y_ageclass	2688309	41.265	5.615	1	42	44	45
raw_splitting	2637102	0.609	0.488	0	0	1	1
raw_bonus_m_edu	1702658	109.425	48.633	0	96	132	504
raw_bonus_m_assist	6687	36.615	32.158	0	12	48	264

Table 1.3: Summary Statistics of the Pension Register's Variables Recoded for NAs

Chapter 2

Methodology and Application

In addition to the statistical description of the variables constituting the Swiss Pension Register (CCO/FSIO) given in chapter 1, we want to analyse our dataset in more details, in order to group similar observations such that we can obtain statistically significant results by applying classification methods. Therefore, clusters should be defined for the Swiss Pension Register (CCO/FSIO).

Milligan (1980) gives a generally accepted definition for the structure of a cluster:

Definition 2.0.1. « Clusters should exhibit the properties of external isolation and internal cohesion. External isolation requires that entities in one cluster should be separated from entities in another cluster by fairly empty areas of space. Internal cohesion requires that entities within the same cluster should be similar to each other, at least within the local metric.» ¹

By carrying out this cluster analysis, we would like to uncover some hidden features from the data. In our case, clusters will help to learn more about the structure of the Swiss Pension Register (CCO/FSIO) such that we will be able to get better results by applying classification methods.

However, we encounter three main difficulties, namely:

1. As not only continuous but also nominal or categorical variables structure the register, we have to choose a clustering method that considers any types of variables;
2. The a priori number of clusters should be in the first step determined, and thus the question of how to fix it is essential;
3. The method should run over big data.

¹This text is taken from Milligan (1980), p. 326.



Given the large size of the Swiss **Pension Register (CCO/FSIO)** and its data heterogeneity, it is impossible to apply the usual clustering methods such as the k -means and the Ward clustering methods.

Therefore, we choose the **KAMILA clustering method** (KAy-means for MlXed LARge data sets) — developed by A. Foss et al. (2016) and implemented in R (cf. A. H. Foss and Markatou (2018b)) — to split our data in several clusters. This method is the best suited for the specificity of the **Pension Register (CCO/FSIO)**.

In the following sections, we formally present **KAMILA** and how we apply it in R on the Swiss **Pension Register (CCO/FSIO)**, in order to obtain the best number of clusters k^* and to respectively split the dataset in several groups according to the retrieved value for the parameter k^* .

2.1 Methodology

The methodology necessary for clustering a mixed-type and large dataset matches the one given by the **KAMILA clustering method** which fulfills the requirement of equitably treating the contribution of categorical and continuous variables in a flexible way, i.e. without strong parametric assumptions.

According to A. Foss et al. (2016)², the **KAMILA clustering method** has the following advantages over the other usual clustering algorithms:

1. The original variables (continuous, categorical or nominal) can be used in their original form and do not need to be transformed in order to have one single variable type. The original information is therefore kept completely intact;
2. The contribution of all variables types is equitably balanced;
3. The clusters are determined in a flexible way, i.e. without any restrictive parametric assumptions, « *generalising the form of the clusters to a broad class of elliptical distributions* »;
4. It is not necessary «*to specify variable weights nor use coding schemes*» in order to apply the algorithm.

²The following arguments are taken from A. Foss et al. (2016), p. 420.

2.1.1 Literature review on mixed-type data

We take up and summarise the explanations of A. Foss et al. (2016) (pp. 420 – 426) in this section.

The most applied strategy to handle mixed-type data is transforming the variables in order to harmonise the data set such that one single data type exists. For instance, it is very common to create new dummy variables for each category of a nominal or ordinal variable. By doing so, the dimensionality of the dataset increases and threatens with multicollinearity problems — among others led by the curse of dimensionality.

Firstly, Hennig and Liao (2013) investigate the mixed-type data handling by applying the dummy transformation of the variables. They compare two different methods, namely the *k*-medoids technique which is based on a dissimilarity-based partitioning approach explained by Kaufman and Rousseeuw (1990) and a model-based clustering approach where mixture components and underlying latent classes define the clusters, namely the *latent class clustering* (LCC) exposed by Vermunt and Magidson (2002).

On one side, Kaufman and Rousseeuw (1990) explain that cluster analysis is the grouping of observations close to another without knowing the form, namely the parameters as the mean and the variance, nor the number of these groups. Representative objects, called medoids, are chosen so that they are good enough to yield the minimal average distance between each of the *k* cluster's medoid and its belonging objects. Hence, this technique of partitioning is called *k*-medoids which can be run by the program PAM (*partitioning around medoids*) also giving a so-called *silhouette plot* ³.

On the other side, Vermunt and Magidson (2002) speak of *latent class* (LC) analysis as being also defined by the explanation given by Kaufman and Rousseeuw (1990) for cluster analysis, where the data belong to *K* latent classes whose form and number are unknown *a priori*.

Moreover, observations sharing the same group are alike with respect to their scores for several variables as they are assumed to come from the same probability distribution whose parameters are yet unknown and to be estimated. However, the LC analysis differs from classical clustering analysis in the fact that LC clustering is based on statistical models related to the sampled population.

Therefore, according to Vermunt and Magidson (2002), LC analysis, as a probabilistic clustering approach, allows to take into account the uncertainty that a certain object belongs to a specific class, which let us think of the conceptual relationship between fuzzy clustering techniques and the LC analysis. While it is possible to classify other objects with LC analysis thanks to the estimated parameters of the statistical model, it is not the case with standard fuzzy cluster techniques where the estimated parameters determine the extent of an object's membership to a specific group ⁴.

³For more details, please consult Kaufman and Rousseeuw (1990) p. 41.

⁴For more information, please consult Vermunt and Magidson (2002) p. 2.

In a nutshell, thanks to the parallels existing between cluster analysis and LC analysis, the latter has become an important tool in the clustering methods.

Secondly, Hennig and Liao (2013) investigate the performance of the CLARA (CLustering LArge applications) algorithm (of the R package `cluster`) by using dummy variables. This program has the same goal as PAM but is designed especially for large data sets.

PAM can only stand for a maximal number of $n = 100$ observations as it stores all pairwise distances in a central memory opposite to CLARA which only stores the actual measurements and is therefore more limited regarding some features than PAM.

A. Foss et al. (2016) find out that the dummy variables — coded following the dummy coding strategy suggested by Hennig and Liao (2013) — and some types of hybrid distance metrics — such as the Gower's distance (Gower (1971)) — have a large and unjustifiable influence on the k -medoids algorithms (such as CLARA).

This is not the case with k -means algorithm using centroids (instead of medoids) which represent the centre location of the clusters opposite to the medoids that are real observed objects. This method uses the squared Euclidean distance as metric between the data and the centroids. There are as many centroids as there are clusters. This explains that the k -means algorithm is more flexible than the k -medoids algorithm — which requires the k medoids to be observed — in the sense that the k -means algorithm is « *robust to various forms of continuous error perturbation* »⁵. However, it is not well suited for categorical variables.

In order to find a way to deal with categorical variables, Huang (1998) presents the k -prototypes algorithm which has the advantage to combine several techniques to handle several data types. Although this method is derived from the k -means algorithm, it is different in the sense that it does not only use the squared Euclidean distance (for continuous variables) but also « *matching distance for categorical variables* ». However, this method needs a weighting criterion to balance between continuous and categorical variables. Therefore, it has the same limitation as the methods using the Gower's distance because a weighting factor is needed.

The only method bringing a solution to weigh effectively between the contribution of the categorical and continuous variables is the one given by Modha and Spangler (2003). They develop an optimal weight between different features of the data by means of a generalisation of the Fisher's discriminant.

Thanks to this optimal feature weighting, it is possible to obtain clusters such that the average within-cluster dispersion is minimised and the average between-cluster dispersion is maximised for all the feature spaces. This weighted distortion measure, namely D^α , is adaptively selected in order to obtain well separated feature spaces. D^α involves the within-class and the between-class covariance matrices resulting in the generalised Fisher ratio. Modha and Spangler (2003) integrate this D^α measure in the k -means algorithm. One drawback is that the number of clusters still has to be chosen before running the algorithm.

⁵For more details, please consult Milligan (1980).

Another possibility would be to use parametric methods which perform quite well with categorical and continuous variables. However, the limitation of these models lies in the fact that they cannot stand a large number of categories nor of categorical variables.

Among non-parametric methods, the **Kernel Density** method cannot handle categorical or mixed-type data. Therefore, a new method has been developed by A. Foss et al. (2016), namely the **KAMILA clustering method**, which can handle large datasets and mixed-type data without making strong assumptions about the weight and the number of clusters nor parametric assumptions.

2.1.2 Kamila Clustering Algorithm

As said above, the motivation to use the **KAMILA clustering method** is driven by the large size of the Swiss **Pension Register (CCO/FSIO)**, by the high number of continuous and categorical variables and by the unknown a priori number of clusters present in this register.

As presented by A. Foss et al. (2016) (pp. 426 – 437), the **KAMILA clustering method** is based on the k -means and extended in order that it would neither be needed to set any weights arbitrarily (like dummy coding) for each type of variables nor to have strong parametric assumptions.

The **KAMILA clustering method** is a blended version of the k -means algorithm and the Gaussian-multinomial mixture models (Hunt and Jorgensen (2011)). A. Foss et al. (2016) have then extended it to very large datasets (Chu et al. (2007), Wolfe, Haghighi, and Klein (2008)). Like k -means, we do not need to make strong parametric assumptions about the continuous variables by using **KAMILA**.

A. Foss et al. (2016) use the *overlap in distributions* concept to speak about how much the clusters densities coming from a mixture distribution overlap. A. Foss et al. (2016) formalise the overlap between two random variables by means of the equation (2.1), i.e.

$$\int_{A_1} f_1(t)dt + a \int_{A_2} f_2(t)dt, \quad (2.1)$$

where the integrals will be replaced by sums for the categorical variables and where

$$A_1 = \{x : f_1(x) < f_2(x)\} \quad \text{and} \quad A_2 = \{x : f_2(x) < f_1(x)\}. \quad (2.2)$$

The first component of the equation (2.1) indicates the overlap area between the clusters 1 and 2, that is, the area under the PDF of cluster 1 over the region A_1 where $f_1(x) < f_2(x)$ and the second component refers to the area under the PDF of cluster 2 over the region A_2 where $f_2(x) < f_1(x)$. The overlap is 0 if the two clusters are completely separated and equal to 1 if they come from the same distribution.

Moreover, unlike k -means, **KAMILA** does not impose any assumptions about the weights of the categorical and the continuous variables. With k -means, the trade-off in the choice of weights for the dummy coding (either weights of 0 – 1.00 or 0 – 3.00 for example) constitutes an important limitation. Namely, a high weight given to categorical variables with a low area overlap in the categorical variables and a high area overlap in the continuous variables performs well in the sense that the **Adjusted Rand Index** is higher than in the case when the area overlap is high in the categorical variables and low in the continuous variables.

These results are reversed when the weight of the dummy coding is low (0 – 1.00). This is due to the fact that the contribution of the categorical variables is more important when higher weights are given to them and when the overlap is lower, meaning that the clusters are well separated.

As mentioned earlier, **KAMILA** overcomes the problem of assigning weights to the categorical variables. The algorithm adaptively adjusts to each level of overlapping in the categorical or in the continuous variables. Therefore, it performs better than the other algorithms in all levels of overlapping.

Formal definition

In the following, we formally present the model and the algorithm developed by A. Foss et al. (2016)⁶. First, let us define some terms.

Let be $\mathbf{V}_1, \mathbf{V}_2, \dots, \mathbf{V}_N$ a random multivariate sample of size N such that $\mathbf{V}_i = (V_{i1}, \dots, V_{ip}, \dots, V_{iP})'$ is a continuous random vector with $\mathbf{V}_i \sim f_{\mathbf{V}}(\mathbf{v}) = \sum_{g=1}^G \pi_g h(\mathbf{v}; \mu_g)$, where G is the number of clusters in the mixture; μ_g is the $P \times 1$ centroid of the g -th cluster of \mathbf{V}_i and π_g is the prior probability of drawing an observation from the g -th population; h is a spherical density function. Then, let be $\mathbf{W}_1, \mathbf{W}_2, \dots, \mathbf{W}_N$ a random multivariate sample of size N of $Q \times 1$ discrete random vectors such that $\mathbf{W}_i = (W_{i1}, \dots, W_{iq}, \dots, W_{iQ})'$, $W_{iq} \in \{1, \dots, L_q\}$. The vector \mathbf{W}_i is a mixture of multinomial random variable, with $\mathbf{W}_i \sim f_{\mathbf{W}}(\mathbf{w}) = \sum_{g=1}^G \pi_g \prod_{q=1}^Q m(w_q, \theta_{gq})$, where $m(w, \theta)$ denotes a multinomial probability mass function with parameter vector θ . It is assumed that W_{iq} and $W_{iq'}$ are conditionally independent given membership $\forall q \neq q'$. Finally, let $\mathbf{X}_1, \dots, \mathbf{X}_i, \dots, \mathbf{X}_N$ denotes the random sample with $\mathbf{W}_i = (\mathbf{V}_i', \mathbf{W}_i')'$, \mathbf{V}_i conditionally independent of \mathbf{W}_i , given population membership.

Let be $\hat{\mu}_g^{(t)}$ and $\hat{\theta}_{gq}^{(t)}$, the state of $\hat{\mu}_g$ and $\hat{\theta}_{gq}$ at the t -th iteration of the algorithm. First, A. Foss et al. (2016) propose to estimate the multivariate density function $f_{\mathbf{V}}(\mathbf{v})$ by a kernel approach. In particular, they proof that

$$f_{\mathbf{V}}(\mathbf{v}) = \frac{f_R(r) \Gamma(\frac{p}{2} + 1)}{p r^{p-1} \pi^{p/2}},$$

⁶The following definitions and equations are from A. Foss et al. (2016), p. 429-433.

where $r = \sqrt{\mathbf{v}'\mathbf{v}}$, $r \in [0, \infty)$, $\mathbf{V} = (V_1, \dots, V_p)'$ a random vector that follows a spherical symmetric distribution with centre at the origin. In the estimation of $f_{\mathbf{V}}(\mathbf{v})$, one has simply to estimate the univariate density function $f_R(r)$, which can be done by a traditional kernel approach. Consider furthermore the following quantities:

$$d_{ig}^{(t)} = \sqrt{\sum_{p=1}^P [\xi_p (v_{ip} - \hat{\mu}_{gp}^{(t)})^2],}$$

where ξ_p is an optional weight;

$$r_i^{(t)} = \min_g (d_{ig}^{(t)});$$

$$\hat{f}_R(r)^{(t)} = \frac{1}{N\delta^{(t)}} \sum_{l=1}^N K\left(\frac{r - r_l^{(t)}}{\delta^{(t)}}\right),$$

where $K(\cdot)$ is a kernel function and $\delta^{(t)}$ is a bandwidth at iteration t ;

$$\ln(c_{ig}^{(t)}) = \sum_{q=1}^Q \xi_q \ln(m(w_{iq}; \hat{\theta}_{gq}^{(t)}),$$

where ξ_q is an optional weight.

In order to assign observation i to population g , we postulate the following objective function:

$$H_i(g)^{(t)} = \ln(\hat{f}_{\mathbf{V}}^{(t)}(d_{ig}^{(t)})) + \ln(c_{ig}^{(t)}).$$

Observation i is assigned to population g , if it maximises $H_i(g)^{(t)}$. At the end of the t -th iteration, $\hat{\mu}$ and $\hat{\theta}_{gq}$ are updated. Let be $\Omega_g^{(t)}$ the set of indices of observations assigned to population g at iteration t . Then,

$$\hat{\mu}_g^{(t+1)} = \frac{1}{|\Omega_g^{(t)}|} \sum_{i \in \Omega_g^{(t)}} \mathbf{v}_i.$$

$$\hat{\theta}_{gq}^{(t+1)} = \frac{1}{|\Omega_g^{(t)}|} \sum_{i \in \Omega_g^{(t)}} \mathbb{1}_{\{w_{iq}=l\}}.$$

Given an initialisation, the partition and estimation steps are repeated until convergence. For each initialisation, we calculate

$$\sum_{i=1}^N \max_g \{H_i^{(\text{final})}(g)\}.$$

The partition that maximises the latter is the output result.

2.1.3 Kamila Clustering Algorithm Formalisation

The algorithm 2.1 developed by A. Foss et al. (2016) shows the procedure to obtain a stable solution for the estimates $\hat{\mu}_g^{(t)}, \hat{\theta}_{gq}^{(t)}$.

Algorithm 1: Kamila Clustering

```

1: for User-specified number of initialisations do
  Initialise  $\hat{\mu}_g^{(0)}, \hat{\theta}_{gq}^{(0)}, \forall g, q$ 

2:   repeat
     PARTITION STEP

3:      $d_{ig}^{(t)} \leftarrow \text{dist}(v_i, \hat{\mu}_g^{(t)})$ 
4:      $r_i^{(t)} \leftarrow \min_g(d_{ig}^{(t)})$ 
5:      $\hat{f}_V^{(t)} \leftarrow \text{RadialKDE}(r^{(t)})$ 
6:      $c_{ig}^{(t)} \leftarrow \hat{\text{Pr}}(\mathbf{w}_i \mid \text{observation } i \in \text{population } g)$ 
7:      $H_i^{(t)}(g) \leftarrow \ln[\hat{f}_V^{(t)}(d_{ig}^{(t)})] + \ln[c_{ig}^{(t)}]$ 
8:     Assign observation  $i$  to population  $\underset{g}{\text{argmax}} H_i^{(t)}(g)$ 

     ESTIMATION STEP

9:     Calculate  $\hat{\mu}_g^{(t+1)}$  and  $\hat{\theta}_{gq}^{(t+1)}$ 

10:   until Convergence

11:   ObjectiveFun  $\leftarrow \sum_{i=1}^N \max_g H_i^{(\text{final})}(g)$ 

12: end for
    Output partition that maximises ObjectiveFun.

```

Figure 2.1: Kamila Clustering Algorithm

2.1.4 Prediction strength algorithm

Several methods and criteria exist to determine the number of clusters. According to A. Foss et al. (2016) (p. 432), they chose the **prediction strength criterion** (Tibshirani and Walther (2005)) because it is very flexible. It had been preferred over other methods like the *gap statistic*, the *BIC* or the *silhouette width* (among others) because they would have required to be adapted to the **KAMILA** method. The following definitions and variables are taken from Tibshirani and Walther (2005), pp. 513 – 517.

The **prediction strength criterion** allows to determine the optimal number of clusters k^* for a specific dataset. According to Tibshirani and Walther (2005), the best number of cluster denoted by k^* yields a prediction strength (plus-minus the standard deviation) above the threshold of 0.8 in order to get well separated clusters (i.e. not overlapping each other).

Let us define by X_{tr} and X_{te} a training data set respectively a test data set; $C(X_{tr}, k)$ denotes a clustering with k clusters of the training data set; $D[C(X_{tr}, k), X_{te}]_{ij'}$ is the ij' -th element of the square D matrix. $D[C(X_{tr}, k), X_{te}]_{ij'} = 1$ if the observations i and i' belong to the same cluster and are therefore called « *co-membership* » (otherwise $D[C(X_{tr}, k), X_{te}]_{ij'} = 0$).

We rewrite the **prediction strength criterion** definition by Tibshirani and Walther (2005) (p. 514) in the expression (2.3), i.e.

$$ps(k) = \min_{1 \leq j \leq k} \frac{1}{n_{kj}(n_{kj} - 1)} \sum_{i \neq i' \in A_{kj}} D[C(X_{tr}, k), X_{te}]_{ij'}, \quad (2.3)$$

where k denotes the number of clusters and $A_{k1}, A_{k2}, \dots, A_{kk}$ are the indices of the test observations in clusters $1, 2, \dots, k$; $n_{k1}, n_{k2}, \dots, n_{kk}$ stand for the number of observations in these clusters.

In a nutshell, with a number of clusters k as a candidate value, the prediction strength calculation procedure occurs in three steps:

1. The test sample is grouped into k clusters;
2. The training sample is grouped into k clusters;
3. We compute, for each cluster, the proportion of the observations which are classified in the same cluster as with the test sample but using the training set centroids.

If the true number of clusters k_0 is known, that is $k = k_0$, the prediction strength will be very high as the training set clusters will predict very accurately the test set clusters.

Tibshirani and Walther (2005) use the minimum rather than the average in the equation (2.3) in order to take into account the case where $k > k_0$, meaning that there are extra clusters in the training sets, leading to a different clustering in the test set. As a consequence, $ps(k)$ will be much smaller.

In practice, it is very difficult to consider the cluster of each observation. Therefore, only the co-memberships of each observation in some cluster are considered in order to compare the training set and the test set clusterings.

If there is no test sample, an r -fold cross-validation is used in order to estimate the prediction strength (2.3), with the r -fold is used for the test set and the $r - 1$ -folds constituting the training set.

It is also possible to estimate the prediction strength for single observations. Tibshirani and Walther (2005) define the individual prediction strength as

$$\text{ps}(i, k) = \frac{1}{|A_k(i)|} \cdot \sum_{i' \in A_k(i)} 1(D[C(X_{tr}, k), X_{te}]_{ii'} = 1), \quad (2.4)$$

where « $A_k(i)$ are the observations indices i' such that $i \neq i'$ and $D[C(X_{tr}, k), X_{te}]_{ii'} = 1$ » with $|A| = \text{card}(A)$ being the cardinality of A .

2.2 Application in R

For applying the **KAMILA clustering method** to our **Pension Register**, we implemented the package `rrclust` (Lettry (2021)) using some of the R packages coming from `tidyverse` and thus applying the `dplyr` grammar (Wickham et al. (2021)). The global workflow of the `rrclust` package is depicted in the figure 2.2. The green ellipses correspond to modules which are functions accepting only a certain class of inputs, i.e. *tibbles*.

We will use this terminology to identify such data frames in the following sections. These *tibbles* can enter the modules individually or in the form of lists of *tibbles*, i.e. «*tidylists*», containing the *tibbles* in a tidy form needed (or not) by the modules. Each module of this workflow is presented in the appendix A.7.

The red rectangles give the name of the *tibbles* which either are inputs or outputs of the modules ⁷. Therefore, they are the outputs of a transformation of the initial *tibbles*.

The blue rectangles depict the top level outputs, such as LOG indicating the run log with the `rrclust` package version, the `dplyr` library version, the date and the time of the code execution.

The arrows indicate the direction of the process. If there are two arrows between an ellipse and a rectangle in both directions, this means that an input has been renamed especially for this module and is given back as an output. This is the case of `FULL_CONT_DF` and `FULL_CATEG_DF` which are renamed *tibbles* of resp. `CONT_DF` and `CATEG_DF` and which contains the outcome variables `aadr` and `monthly_rent` opposite to their siblings ⁸.

⁷Note that the initial tibble `IND_YEARLY_RR` is an extract of the **Pension Register** for the year 2020 (stored in a `.sas7bdat` format and converted to a `.rds` file), saved in the form of a tidy data frame and exported as a `.csv` file. All raw inputs must be exported into `.csv` files containing tidy data frames, such that they can be read by the modules.

⁸This distinction is explained in the subsection 2.2.2 below.

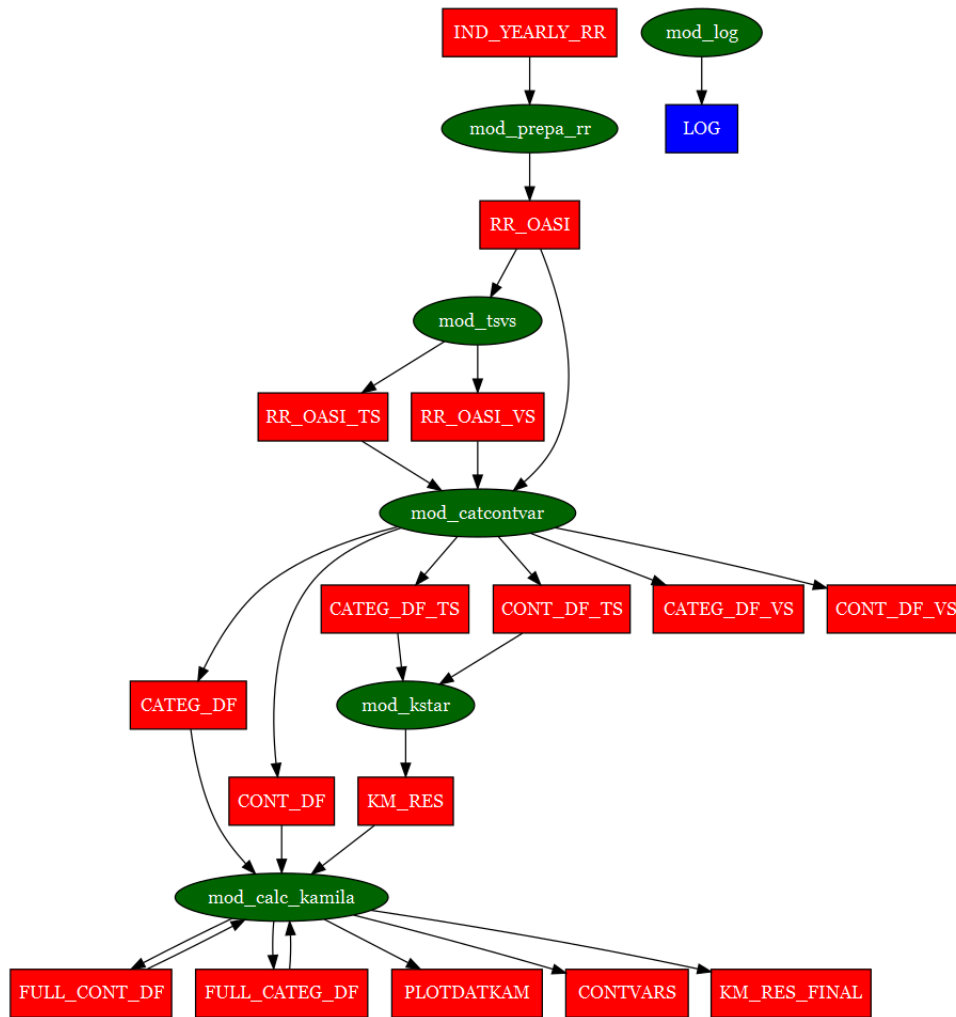


Figure 2.2: Workflow of the package rrclust

2.2.1 Preparation of the data

In a first step, we prepare the data by creating the tibble `RR_OASI` in the module `mod_prepa_rr`⁹. The initial extract of the *Pension Register* 2020, namely `IND_YEARLY_RR`, is treated by the module `mod_prepa_rr` where the final variables' names are defined and stored in `RR_OASI` (cf. table 1.2).

Then, a training set and a validation set are produced in the module `mod_tsvs`¹⁰. In our case, the validation set is the same as the test set, since we do not perform any k -fold cross-validation. The resulting *tibbles*, respectively `RR_OASI_TS` and `RR_OASI_VS`, contain less observations than the initial `RR_OASI` in order to be able to run the *KAMILA* algorithm.

⁹Please consult the script B.3.

¹⁰Please consult the script B.4.

Otherwise, the algorithm for finding the best number of clusters k^* cannot be executed due to the too large dataset and the memory limit of R. Therefore, the training set `RR_OASI_TS` contains 0.1% of the total number of observations, i.e. $2\,688\,607/1000 = 2688$. This training set contains already a sufficient number of observations in order to obtain good results.

The validation set contains the remaining observations, that is 2 685 919 individuals. However, the validation set will not be used as it is impossible to execute the algorithm in order to find k^* with so many observations because of the too large time needed for the computation and the memory limit. Therefore, once the k^* is found by using the training set, this parameter will simply enter the final clustering algorithm applied to the whole initial dataset constituted by the initial 2 688 607 observations. The *tibbles* `RR_OASI`, `RR_OASI_TS` and `RR_OASI_VS` are then split into two subsets of categorical and continuous variables used in the algorithm execution.

2.2.2 Determination of k^*

The subsets of the training set `RR_OASI_TS`, namely `CATEG_DF_TS` and `CONT_DF_TS`, are used to find the parameter k^* . The **KAMILA** algorithm is executed using `CATEG_DF_TS` and `CONT_DF_TS` containing respectively the categorical variables listed in the table 2.1 and the continuous variables in the table 2.2. The outcome variables `aadr` and `monthly_rent` are excluded from the `CONT_DF_TS` as the determination of the best number of clusters must not depend on them.

The parameter k^* is computed by means of the function `kamila` belonging to the `kamila` package¹¹. The necessary parameters to run this function are defined in the script B.7 as well as in the parameters files C.1 and C.2. The k^* estimation occurs in the module `modkstar`¹². The resulting k^* is represented in the figure 3.1.

Variable	Definition	Values
<code>nat</code>	Nationality (dummy)	1 if foreign, 0 if Swiss;
<code>resid</code>	Residence (dummy)	1 if foreign country, 0 if Switzerland;
<code>sex</code>	Sex (dummy)	1 if female, 0 if male;
<code>marital_stat1</code>	Marital status for divorced (dummy)	1 if divorced, 0 otherwise;
<code>marital_stat3</code>	Marital status for married (dummy)	1 if married, 0 otherwise;
<code>marital_stat4</code>	Marital status for widowed (dummy)	1 if widowed, 0 otherwise;
<code>benef_type1</code>	Old-age (dummy)	1 if old-age, 0 otherwise.

Table 2.1: Definition of the used categorical variables

¹¹ Please consult the package documentation A. H. Foss and Markatou (2018b) and the [CRAN reference manual](#).

¹² Please consult the script B.7.

Variable	Definition	Values
age	Age (numeric)	{0, 1, ..., 98, 99};
age_retire	Observed retirement age (numeric)	{62, 63, ..., 69, 70}, NA for survivor insurance beneficiaries;
scale	Factor relative to the number of contribution years to the OASI (numeric)	{1, 2, ..., 43, 44}.

Table 2.2: Definition of the used continuous variables

2.2.3 Splitting the Pension Register into k^* clusters

After having estimated the parameter k^* defining the best number of clusters according to the procedure explained in the section 2.2.2 with the module written in the script B.7, the KAMILA algorithm is then run with the newly estimated k^* in order to split the initial Pension Register into this optimal number of clusters given by k^* . This occurs as written in the script B.8.

This splitting procedure is applied to the initial *tibbles* CONT_DF and CATEG_DF which respectively contain all continuous variables except for the outcome variables aadr and monthly_rent and all categorical variables except for the nominal ones called marital_stat and benef_type.

The results of the clustering are summarised in the *tibbles* KM_RES, KM_RES_FINAL and PLOTDATKAM. The FULL_CONT_DF and FULL_CCATEG_DF are the same *tibbles* as CONT_DF and CATEG_DF except that they respectively contain the outcome variables aadr and monthly_rent and the marital_stat and benef_type.

Chapter 3

Results

In this chapter, we document our results and describe the composition of the clusters obtained with our R package *rrclust* (Lettry (2021)).

3.1 Kamila estimated parameters

3.1.1 Best number of cluster k^*

In order to find the best number of clusters and the best set of variables which allowed us to get an optimally clustered Swiss Pension Register (CCO/FSIO), we ran the KAMILA algorithm for several combinations of variables. We present the results in the table 3.1.

These sets of variables differ from each other in the way that some contain the original nominal variables from the Swiss Pension Register (CCO/FSIO) and others the dummy variables which stem from the nominal ones. For example, the dummy variables `marital_stat1`, `marital_stat2`, `marital_stat3` and `marital_stat4` are derived from the four categories of the nominal variable `marital_stat`, as described in the table 1.1.

We then compared the log-likelihoods and the prediction strengths with a threshold set at 0.8 for all runs. We found that the best log-likelihood and the best prediction strength difference are obtained with the combination of the variables highlighted in light blue in the first row of the table 3.1. We presented these variables in the tables 2.1 and 2.2.

Thus, the optimal clustering is obtained with the set of variables highlighted in light blue in the table table 3.1. We qualify it as optimal because it allows to get the maximal number of clusters (5) with the best results in terms of two criteria, that is :

1. the sharp split in the prediction strength values between 5 and 6 clusters;
2. the highest log-likelihood value.

We tend to prefer a higher number of clusters in order to be able to better understand the PR (CCO/FSIO) and to be sure not to forget any particularities of this dataset.



Categorical explanatory variables			Continuous explanatory variables			kstar	Log Likelihood	Pred. Strength of kstar	Pred. Strength of (kstar + 1)	Difference Pred. Strength
sex_nat, resid, benef_type1, marital_stat1, marital_stat3, marital_stat4			age, age_retire, scale			5	-3754564.13	0.8667	0.5029	0.3638
sex_nat, resid, benef_type, marital_stat, splitting			age, age_retire, scale, bonus_m_assist			4	-4435062.29	0.8911	0.7210	0.1701
sex_nat, resid, benef_type1, marital_stat, splitting			age, age_retire, scale, bonus_m_assist, contrib_m_ind			4	-4497168.95	0.8188	0.5693	0.2495
sex_nat, resid, benef_type1, marital_stat1, marital_stat3, marital_stat4, splitting			age, age_retire, scale			5	-4551460.20	0.9073	0.6479	0.2595
sex_nat, resid, benef_type, marital_stat, splitting			age, age_retire, scale, bonus_m_edu			4	-4700452.65	0.9104	0.7303	0.1801
sex_nat, resid, benef_type1, marital_stat1, marital_stat3, marital_stat4			age, age_retire, scale, bonus_m_edu, bonus_m_assist			3	-4981013.66	0.9348	0.6103	0.3245
sex_nat, resid, benef_type, marital_stat			age, age_retire, scale, bonus_m_edu			3	-5094138.69	0.8940	0.6783	0.2158
sex_nat, resid, benef_type1, marital_stat1, marital_stat3, marital_stat4			age, age_retire, scale, contrib_m_ind, bonus_m_edu, bonus_m_assist			4	-5244529.08	0.8981	0.5702	0.3279
sex_nat, resid, benef_type1, marital_stat1, marital_stat3, marital_stat4			age, age_retire, scale, bonus_m_assist			3	-5337479.66	0.8835	0.7349	0.1486
sex_nat, resid, benef_type1, marital_stat1, marital_stat3, marital_stat4			age, age_retire, scale, bonus_m_edu			3	-5440738.46	0.8876	0.6106	0.2770
sex_nat, resid, benef_type1, marital_stat1, marital_stat3, marital_stat4, splitting			age, age_retire, scale			5	-5547647.72	0.8177	0.6100	0.2077
sex_nat, resid, benef_type1, marital_stat1, marital_stat3, marital_stat4, splitting			age, age_retire, scale, contrib_m_ind			3	-5641011.53	0.9430	0.7371	0.2058
sex_nat, resid, benef_type1, marital_stat1, marital_stat3, marital_stat4, splitting			age, age_retire, scale, contrib_m_ind			3	-5723097.28	0.9467	0.7779	0.1687
sex_nat, resid, benef_type1, marital_stat1, marital_stat3, marital_stat4, splitting			age, age_retire, scale			5	-5754665.75	0.8201	0.5259	0.2943
sex_nat, resid, benef_type1, marital_stat1, marital_stat3, marital_stat4, splitting, capping			age, age_retire, scale			4	-5838337.73	0.9172	0.6494	0.2678
sex_nat, resid, benef_type1, marital_stat1, marital_stat3, marital_stat4, splitting			age, age_retire, scale, contrib_m_ind			3	-5893652.04	0.9450	0.7604	0.1845
sex_nat, resid, benef_type, marital_stat, capping			age, age_retire, scale, contrib_m_ind, bonus_m_edu, bonus_m_assist			3	-6024282.06	0.9356	0.7459	0.1897
sex_nat, resid, benef_type, marital_stat, splitting			age, age_retire, scale			4	-6063353.48	0.8950	0.7337	0.1613
sex_nat, resid, benef_type, marital_stat, splitting			age, age_retire, scale			5	-6098213.31	0.8650	0.5787	0.2863
sex_nat, resid, benef_type, marital_stat, splitting			age, age_retire, scale, bonus_m_assist			4	-6186716.21	0.8241	0.6899	0.1342
sex_nat, resid, benef_type, marital_stat, splitting			age, age_retire, scale, bonus_m_edu			4	-6272768.85	0.8595	0.6262	0.2333
sex_nat, resid, benef_type, marital_stat, splitting			age, age_retire, scale, bonus_m_assist			3	-6287347.47	0.8837	0.7621	0.1216
sex_nat, resid, benef_type, marital_stat			age, age_retire, scale, bonus_m_assist			3	-6549711.55	0.8418	0.6876	0.1542
sex_nat, resid, benef_type, marital_stat			age, age_retire, scale			3	-6630621.24	0.9703	0.7645	0.2058
sex_nat, resid, benef_type1, marital_stat, splitting			age, age_retire, scale, bonus_m_edu			3	-6653422.12	0.8584	0.7264	0.1320
sex_nat, resid, benef_type1, marital_stat			age, age_retire, scale, contrib_m_ind			3	-6847668.70	0.9443	0.7127	0.2316
sex_nat, resid, benef_type1, marital_stat			age, age_retire, scale, bonus_m_assist			3	-6936411.96	0.9454	0.7991	0.1463
sex_nat, resid, benef_type1, marital_stat			age, age_retire, scale, contrib_m_ind			4	-6968177.82	0.8762	0.6335	0.2427
sex_nat, resid, benef_type1, marital_stat			age, age_retire, scale, bonus_m_edu			3	-7059641.76	0.9493	0.6480	0.3013
sex_nat, resid, benef_type1, marital_stat			age, age_retire, scale			4	-7065454.09	0.9181	0.6812	0.2369
sex_nat, resid, benef_type1, marital_stat			age, age_retire, scale, bonus_m_assist, contrib_m_ind			3	-7132059.06	0.9281	0.6936	0.2345
sex_nat, resid, benef_type, marital_stat1, marital_stat3, marital_stat4, splitting			age, age_retire, scale, bonus_m_edu, contrib_m_ind			3	-7192353.69	0.9318	0.7745	0.1573
sex_nat, resid, benef_type, marital_stat1, marital_stat3, marital_stat4			age, age_retire, scale, contrib_m_ind, bonus_m_edu, bonus_m_assist			3	-7367326.63	0.9601	0.6877	0.2724
sex_nat, resid, benef_type, marital_stat1, marital_stat3, marital_stat4			age, age_retire, scale, contrib_m_ind			3	-7508686.13	0.9552	0.6572	0.2979
sex_nat, resid, benef_type, marital_stat1, marital_stat3, marital_stat4			age, age_retire, scale, contrib_m_ind, bonus_m_edu, bonus_m_assist			3	-7629690.62	0.9516	0.6980	0.2535
sex_nat, resid, benef_type, marital_stat1, marital_stat3, marital_stat4, splitting			age, age_retire, scale, bonus_m_edu, bonus_m_assist			2	-7725684.88	0.9524	0.7787	0.1737
sex_nat, resid, benef_type, marital_stat1, marital_stat3, marital_stat4, splitting			age, age_retire, scale, contrib_m_ind			3	-8453709.88	0.9548	0.7789	0.1760
sex_nat, resid, benef_type, marital_stat1, marital_stat3, marital_stat4			age, age_retire, scale, contrib_m_ind			3	-8453709.88	0.9548	0.7789	0.1760
sex_nat, resid, benef_type, marital_stat1, marital_stat3, marital_stat4			age, age_retire, scale, contrib_m_ind, bonus_m_edu, bonus_m_assist			3	-8578923.58	0.8839	0.7848	0.0991
sex_nat, resid, benef_type, marital_stat1, marital_stat3, marital_stat4			age, age_retire, scale, bonus_m_edu, bonus_m_assist			2	-9368986.15	0.9624	0.7607	0.2017

Table 3.1: Log-likelihood and Prediction Strength analysis of explanatory variables sets

The values of the **prediction strength criterion** presented in the table 3.1 are depicted by the graph 3.1. For example, the value of the **prediction strength criterion** for the best combination of variables (cf. row highlighted in light blue) is 0.8667 for $k^* = 5$ and 0.5029 for $k = k^* + 1 = 5 + 1 = 6$.

Thus, the difference of 0.3638 between these prediction strength values is the largest ($0.8667 - 0.5029 = 0.3638$) among all combinations of variables between k^* and $k = k^* + 1$, which means that the combination of variables shown in the highlighted row leads to well separated clusters and to a very clear k^* value being 5.

This sharp split between $k^* = 5$ and $k = k^* + 1 = 6$ is clearly shown by the figure 3.1.

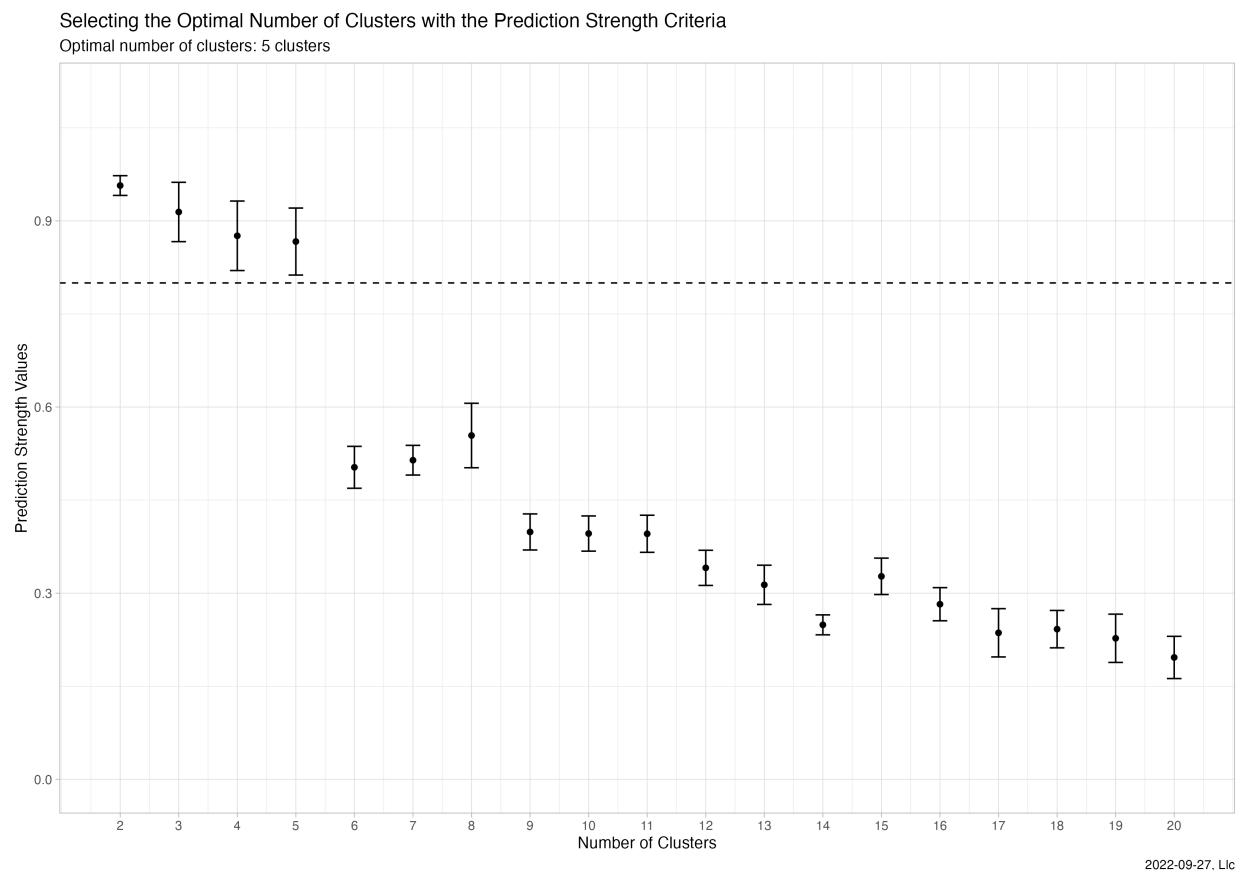


Figure 3.1: Prediction Strength Values for Determining the Optimal Number of Clusters k^*

We thus selected the largest number of clusters whose prediction strength is above 0.8. This threshold is estimated as sufficient for determining a reliable parameter k^* , describing the best number of clusters.

3.1.2 Characteristics of the clusters

After having run the **KAMILA clustering method** on the whole Swiss **Pension Register (CCO / FSIO)** using our best combination of variables (cf. section 3.1.1) as explanatory variables with the parameter $k^* = 5$ according the explanations given in the section 2.2.3, we could produce some tables to analyse the clusters features.

We thus construct the tables 3.2 and 3.3 based on the contingency table A.1. As the first table describes the composition of each cluster, the second one documents which segment of the population constitutes the major part of each cluster. The table 3.3 is mainly a summary of the contingency table A.1 and gives a good overview of the main groups constituting the Swiss **Pension Register (CCO/FSIO)** for the year 2020. The summary statistics of each cluster are presented in the tables of the appendix A.7.

Cluster	Sex	Type of Benefit	Marital Status	Residence	Nationality	Particularities
1	All	Old-age	D/S/W	All	All	None;
2	All	Old-age	M	All	All	None;
3	All	Other	All	Foreign	Foreign	Special case;
4	All	Old-age	All	Foreign	Foreign	None;
5	All	Other	All	All	All	Special case.

Source: Estimation based on the Swiss **Pension Register (CCO/FSIO)** for the year 2020 and on the clustering results obtained with $k^* = 5$. This table is based on the contingency table A.1.

Table 3.2: Clusters Characteristics

We should note a first special case in the table 3.2 showing that the majority of the individuals in the cluster 3 has a foreign nationality and lives in a foreign country. However, married individuals with the Swiss nationality in a foreign country (male or female) or living in Switzerland with a foreign nationality (female) are also part of cluster 3.

Then, we should note a second special case in the table 3.2 because there are more individuals who get a type of pension different from old-age in cluster 3 than in cluster 5 for the following categories: Swiss married individuals living in a foreign country (male or female), foreign married females living in Switzerland, and foreign individuals (male or female) living in a foreign country.

The major part of the clusters can be easily summed up as there is a clear partition of the observations according to the variables concerning the sex, the type of benefit, the marital status, the nationality and the residence. However, the clusters 3 and 5 are related to each other with respect to a particular marital status value, namely married, for

which it is not clear which cluster it belongs to. They exclusively concern the individuals who get a type of pension different from old-age among which Swiss married females and males living in Switzerland take clearly part of the cluster 5, whereas the other married groups are mixed up between cluster 3 and 5.

Therefore, it must be another variable which can explain this discrepancy, namely, the continuous variable `scale`. The graphs [A.30a](#) and [A.31a](#) show a clear partition between both clusters 3 and 5. As the female individuals who get a type of pension different from old-age seem to be relatively well separated in two groups (cf. graph [A.31a](#)), the graph [A.30a](#) shows that the number of divorced and married male individuals who get a type of pension different from old-age is very low as only a few individuals appear to be in the respective divorced and married categories. Therefore, the distribution of the male observations between the clusters 3 and 5 is very questionable for both of these marital status among the individuals who get a type of pension different from old-age.

The table [3.3](#) is the most important result of our application of the [KAMILA](#) method on the Swiss [Pension Register \(CCO/FSIO\)](#). It classifies the observations in the five clusters found with the [KAMILA clustering method](#). It gives the reverse view of the one given by the table [3.2](#) and simplifies the results analysis. Moreover, the table [3.3](#) allows to consider the Swiss [Pension Register \(CCO/FSIO\)](#) in terms of five groups constituting the [OASI](#) pensions. As it would have been unreadable to classify the individuals according to the three other continuous variables used to determine the k^* parameter — namely `age_retire`, `scale` and `age` —, the table [3.3](#) summarises the result of the marginal sums relative to the categorical variables listed in the table [2.1](#) according to the contingency table [A.1](#).

		Swiss								Foreign							
		Switzerland				Foreign Country				Switzerland				Foreign Country			
		D	S	M	W	D	S	M	W	D	S	M	W	D	S	M	W
Female	Old-age	C1	C1	C2	C1	C1	C1	C2	C1	C1	C1	C2	C1	C4	C4	C4	C4
	Other	C5	C5	C5	C5	C5	C5	C3	C5	C5	C5	C3	C5	C3	C3	C3	C3
Male	Old-age	C1	C1	C2	C1	C1	C1	C2	C1	C1	C1	C2	C1	C4	C4	C4	C4
	Other	C5	C5	C5	C5	C5	C5	C3	C5	C5	C5	C5	C5	C3	C3	C3	C3

As only a few married male individuals who get a type of pension different from old-age are observed in this category (2 only, cf. table [A.1](#)), it is hard to assess whether this group is part of cluster 5 or cluster 3. The same prevails for the divorced ones, but as it seems to be congruent with the female classification, we assume there is no exception for the males.

Table 3.3: Clusters Distribution

Regarding the continuous variables presented in the table [2.2](#), the number of contribution years to the [OASI](#) (represented by the variable `scale`) has a clear impact on the classification of the observations into the clusters (cf. figures [A.25](#), [A.26](#), [A.30](#), [A.31](#)). On the contrary, the age and the age of effective retirement do not have any role to play in the repartition of the observations in the clusters, as the individuals from all clusters are counted in all the registered ages (cf. figures [A.23](#), [A.24](#), [A.27](#), [A.28](#), [A.29](#), [A.32](#)). However, one can see thanks

to the graph A.32 that the individuals belonging to the clusters 1 and 2 tend to have a higher monthly pension amount than those in the cluster 4, for all retirement ages. More particularly, the cluster 4 is constituted by old-age pensioners whose rent has a larger variance ($\sqrt{s^2} = 430.957$) than those from clusters 1 ($\sqrt{s^2} = 352.59$) and 2 ($\sqrt{s^2} = 250.771$).¹

As said earlier, according to the figures A.25, A.26, A.30 and A.31, the number of contribution years to the OASI affects the clusters' definition in the following sense:

- Clusters 3 and 5: as we could not clearly say with the help of the table A.1 how exactly the individuals are distributed in the clusters 3 and 5, the `scale` variable helps to clarify this point. The individuals who get a type of pension different from old-age belonging to the cluster 3 tend to have paid the OASI contribution during less years than those classified in the cluster 5. The threshold in the `scale` variable between the clusters 3 and 5 varies in function of the marital status, the nationality and the residence.
- Clusters 1, 2 and 4: in addition to the characterisation of these clusters in the table 3.3, it is clear that the clusters 1 and 2 include the old-age pensioners whose number of contribution years to the OASI is higher than those belonging to the cluster 4. The threshold in the `scale` variable between the clusters 4 and 1 resp. 2 varies in function of the marital status, the nationality and the residence.

3.2 Distribution of the AADR among the clusters

The graphs of the section A.3 show the distribution of the natural logarithm of the AADR among the clusters and the categories of the variables from the table 2.2. The values of the corresponding skewness and kurtosis for each of the whole clusters are presented in the table 3.4

The figures 3.2, 3.3 and 3.4 show, respectively, the histograms, the density function and the empirical cumulative distribution function of the natural logarithm of AADR and of the monthly pension amount. Besides, the figure 3.5 presents the normal quantile-quantile plot of the AADR standardised for range and of the monthly pension amount.

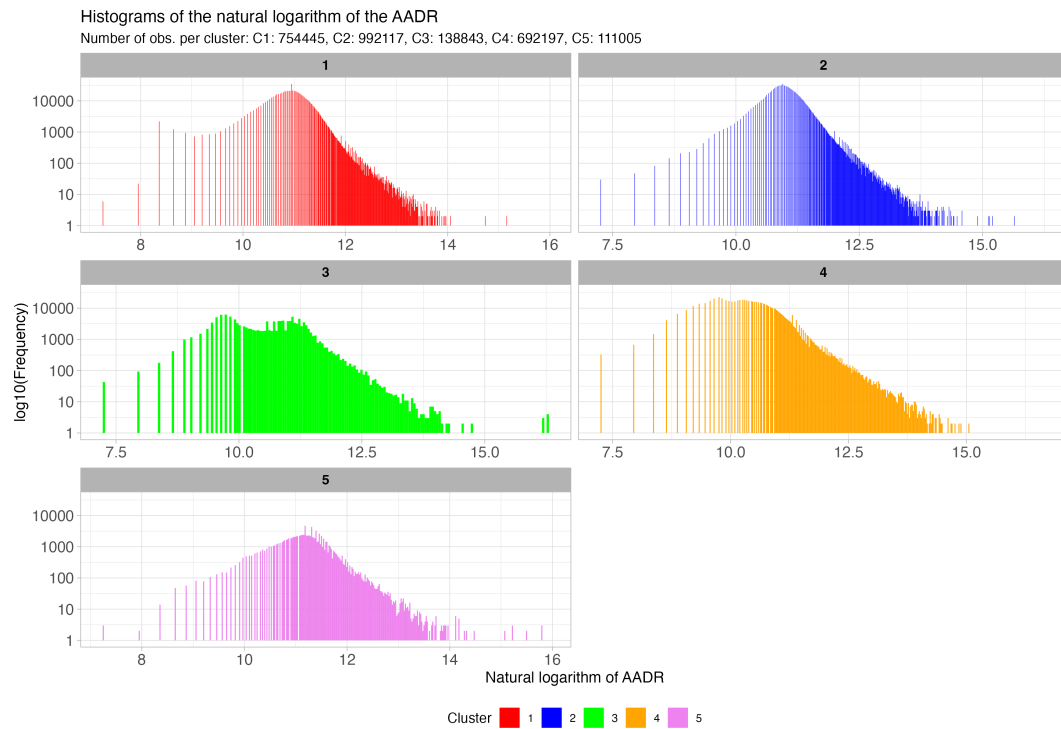
¹The variance values are taken from the respective tables A.5 for the cluster 4, A.2 for the cluster 1 and A.3 for the cluster 2.

	Moment	Cluster	AADR (CHF)	Monthly Pension Amount (CHF)
1	Skewness	1	37.30	-1.34
2	Skewness	2	51.13	-0.59
3	Skewness	3	84.20	1.41
4	Skewness	4	92.32	1.38
5	Skewness	5	44.07	-0.09
6	Kurtosis	1	5155.25	4.93
7	Kurtosis	2	7735.97	7.29
8	Kurtosis	3	9600.20	4.14
9	Kurtosis	4	22398.85	4.13
10	Kurtosis	5	3101.53	1.57

Table 3.4: Table of Moments pro Cluster

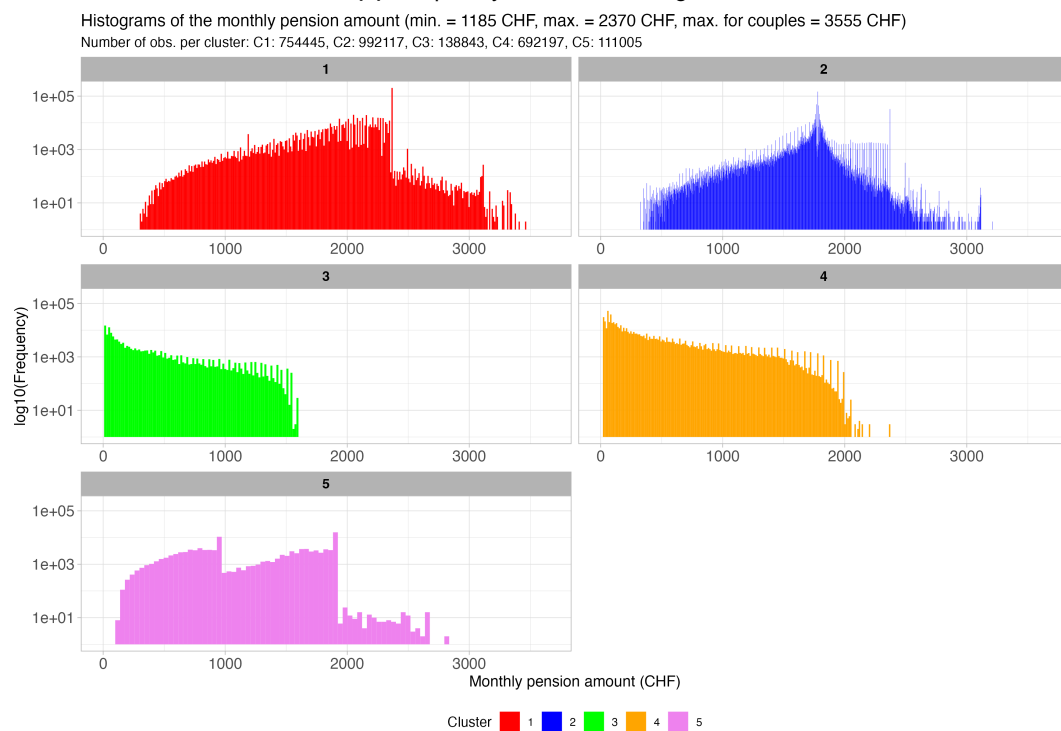
We can summarise the information given by these graphs and by the table 3.4 by the following observations:

- The natural logarithm of the AADR tends to follow a right-skewed and peaky distribution in all clusters. The distribution in the cluster 3 appears to be bimodal;
- The monthly pension amount tends to follow a left-skewed distribution for the clusters 1, 2 and 5 and a right-skewed distribution for the clusters 3 and 4. All clusters apart from the cluster 5 have a high value for the kurtosis, meaning that they have a peaky distribution. The cluster 5 have a kurtosis value equal to 1.57 and is thus flatter than the other clusters' distributions. We can notice that the clusters 2 and 5 have a bimodal distribution;
- According to the normal quantile-quantile plots in the figure 3.5, the AADR (standardised for range) follows a normal distribution apart from the deviating tail. For the monthly pension amount, only its distribution in the cluster 1 appears to follow more or less a normal distribution. The other clusters (2, 3, 4 and 5) have a distribution which extremely differs from the normal one.



2022-09-27, Lic

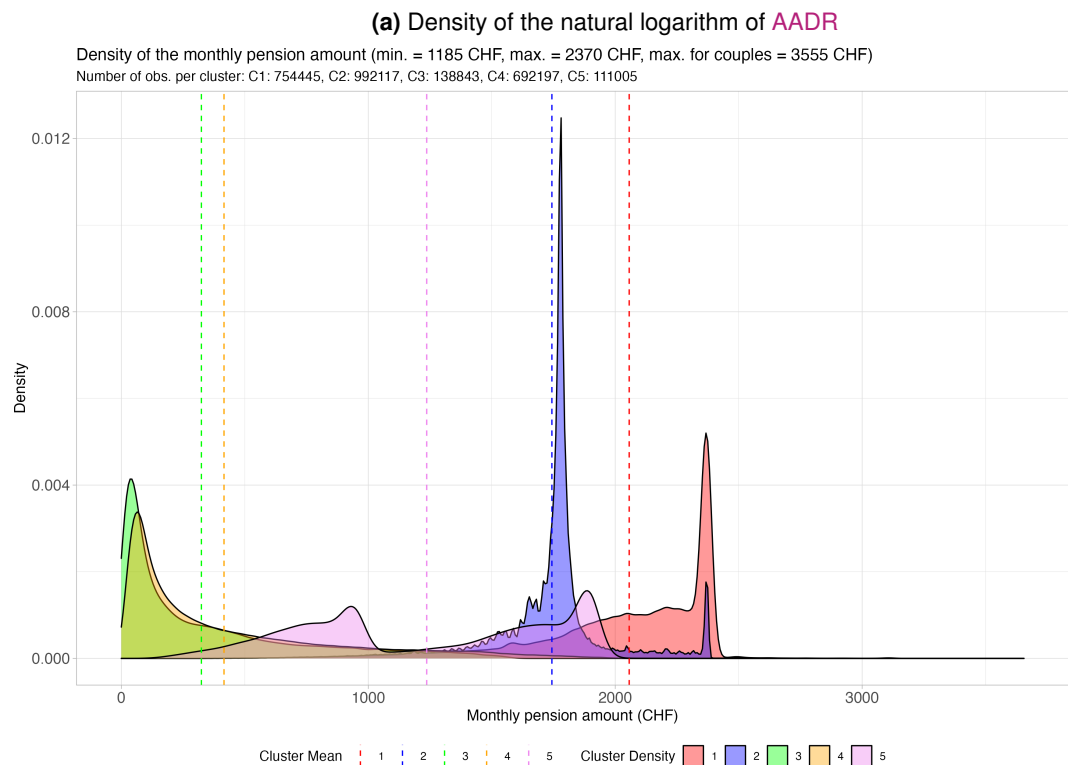
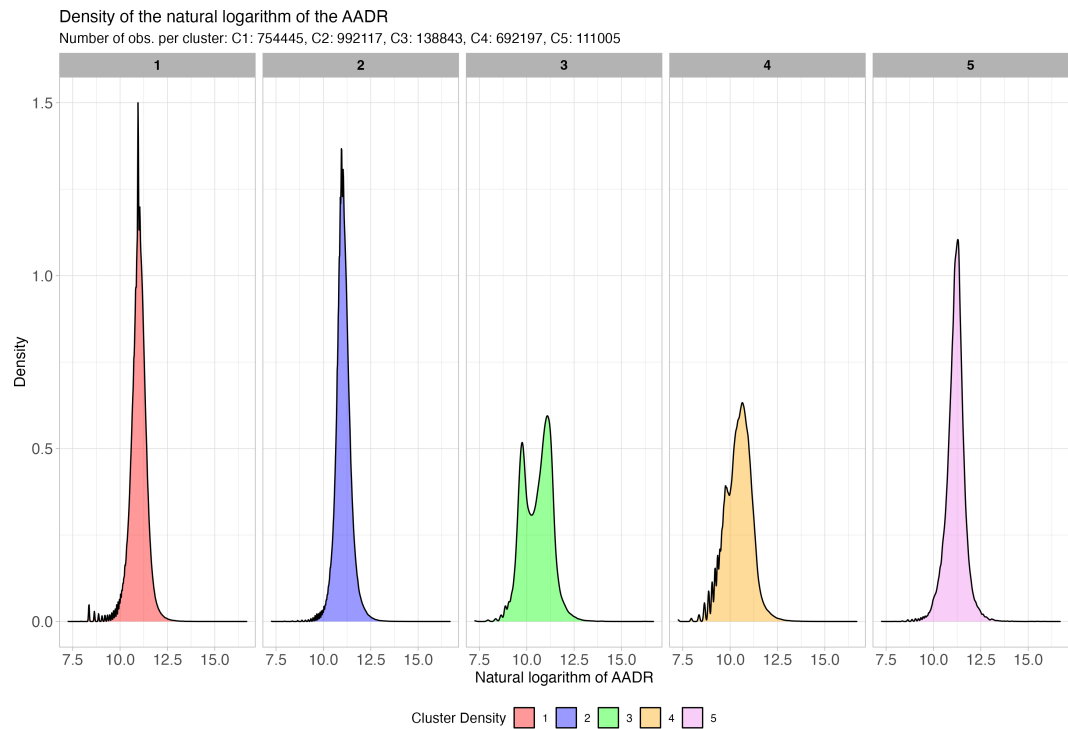
(a) Frequency of the natural logarithm of AADR



2022-09-27, Lic

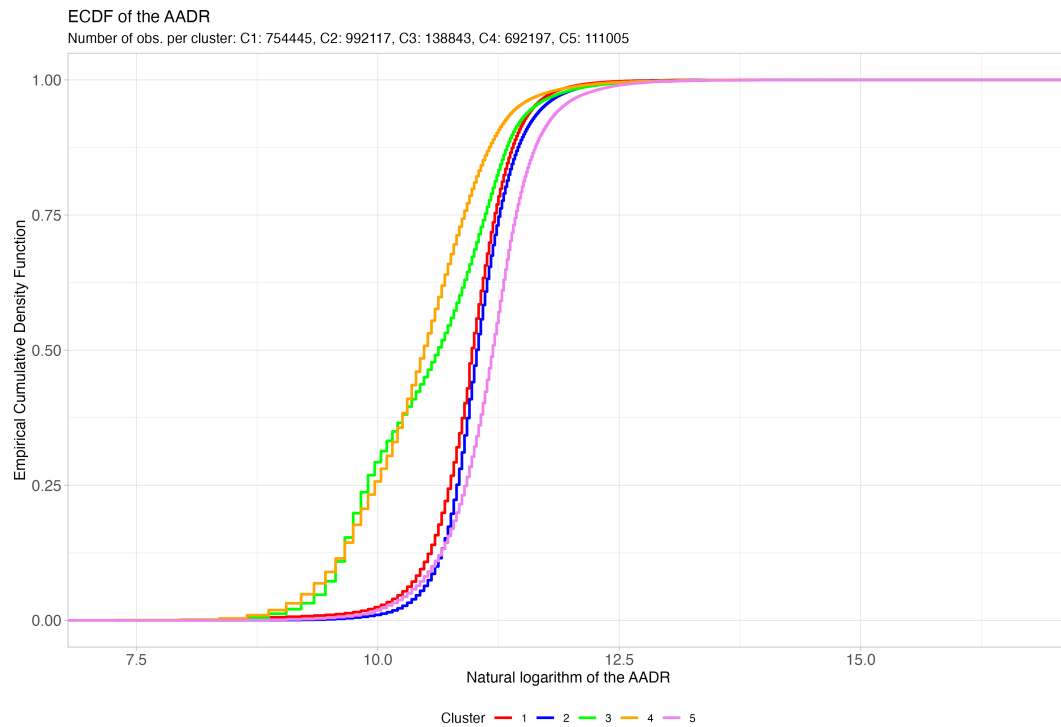
(b) Frequency of the monthly pension amount

Figure 3.2: Histograms of the AADR and of the monthly pension amount



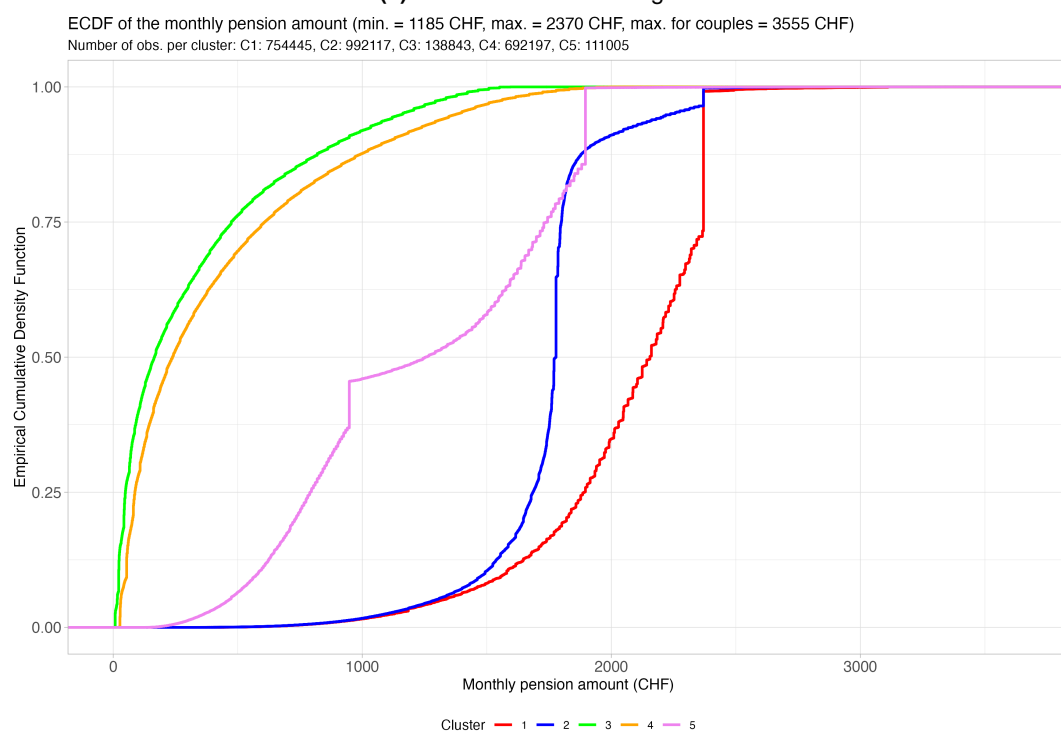
(b) Density of the monthly pension amount

Figure 3.3: Density of the AADR and of the monthly pension amount



2022-09-27, Lic

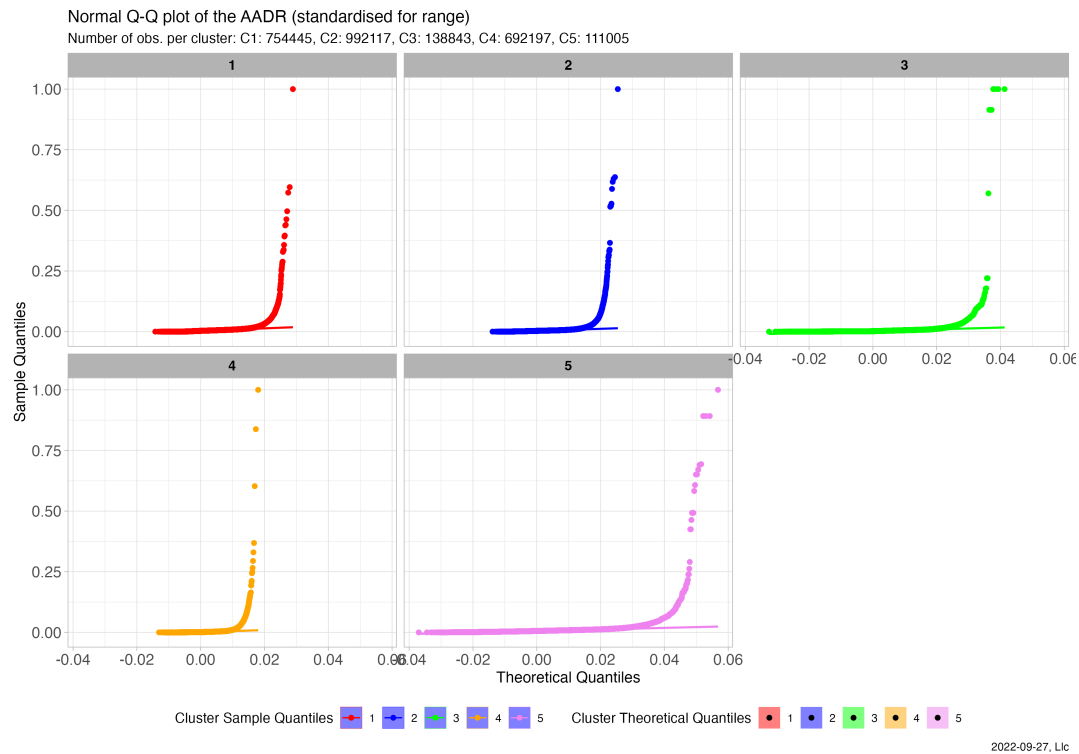
(a) ECDF of the natural logarithm of AADR



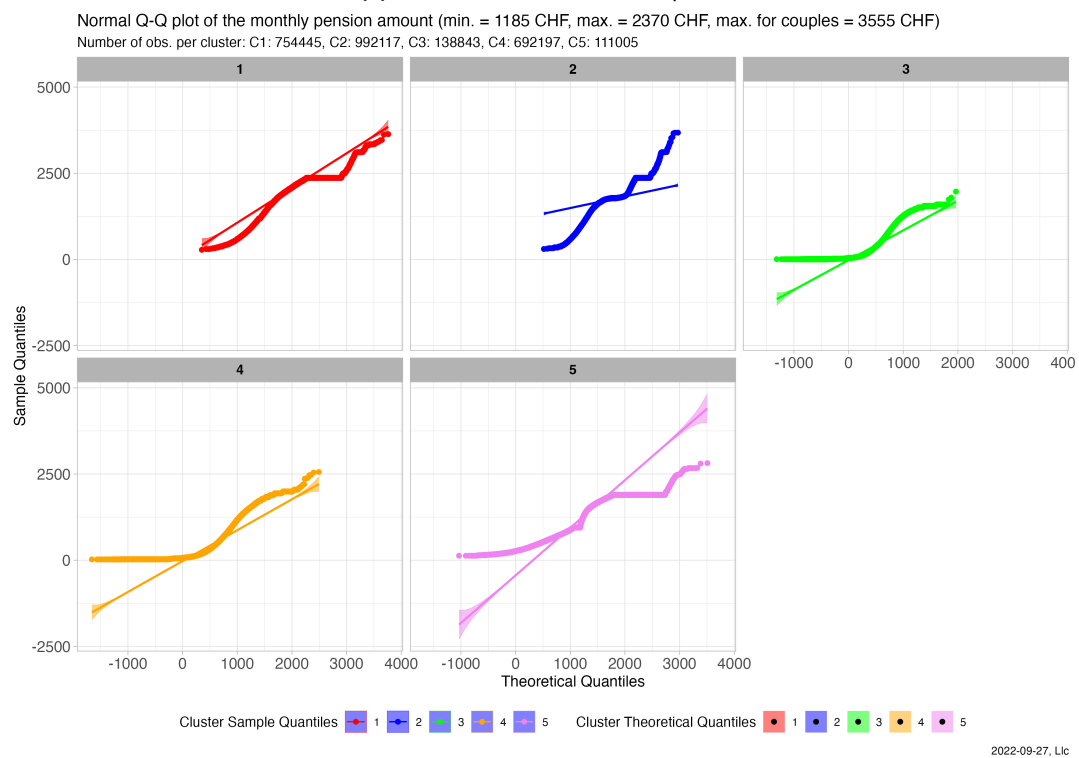
2022-09-27, Lic

(b) ECDF of the monthly pension amount

Figure 3.4: Empirical Cumulative Distribution Function of the AADR and of the monthly pension amount



(a) Normal Quantile-Quantile plot of the AADR



(b) Normal Quantile-Quantile plot of the monthly pension amount

Figure 3.5: Normal Quantile-Quantile plots of the AADR and of the monthly pension amount

Conclusion

This working paper can be considered as a contribution to the application of the **KAMILA clustering method** and to the global knowledge of the **OASI** register data.

This is the first time that such an instrument has been developed for the **FSIO**. It allows classifying the observations of the Swiss **Pension Register (CCO/FSIO)** into groups according to sociodemographic and economic characteristics. This has a practical interest since we can now consider these data as 5 groups of individuals and better understand them.

These clusters can be compared to each other and viewed in a contingency table according to the marital status, the nationality, the residence, the type of pension and the sex of the **OASI** pensioners. The contingency table **A.1** shows the number of individuals in each of the 5 clusters thanks to these categorical explanatory variables. The table **3.3** summarizes the classification of the table **A.1** in the sense that each cell only shows the cluster with the highest number of occurrences in each category.

Three other continuous explanatory variables have entered in the **KAMILA** algorithm in order to determine the best number of clusters for this Swiss **Pension Register (CCO/FSIO)**, namely the scale (i.e. the number of years of contributions payments), the age and the retirement age of the **OASI** pensioners. These variables cannot enter into the contingency table but can be evaluated in the graphs of the appendices **A.5** and **A.6**. For example for the male pensioners in the graphs **A.30a** and **A.30b**, we can see a clear difference of the scale distribution between the clusters while considering the monthly pension amount distribution.

The application of the **KAMILA clustering method** occurred in the R package `rrclust` (Lettry (2021)) which was written following the `dplyr` grammar and thus working with some of the R packages coming from `tidyverse`. The workflow of this package described in the figure **2.2** reveals a flow of *tidy* data frames, namely *tibbles*, coming in and out of computational modules as depicted by the blue and red rectangles as well as by the ellipses.

The final objective of this working paper was to characterise the distributions of the **Annual Average Determinant Revenue** and of the **OASI** monthly pension amount among the clusters. These distributions have been drawn in the section **3.2**. According to these graphs, we drew some conclusions about the shape of the curves which we described according to the skewness and kurtosis as well as by means of normal quantile-quantile plots.



Annexes



This appendix contains graphs, histograms, descriptive statistics as well as summary statistics for each cluster separately.

A.1 OASI monthly pension amount in function to AADR (pension system 2021)

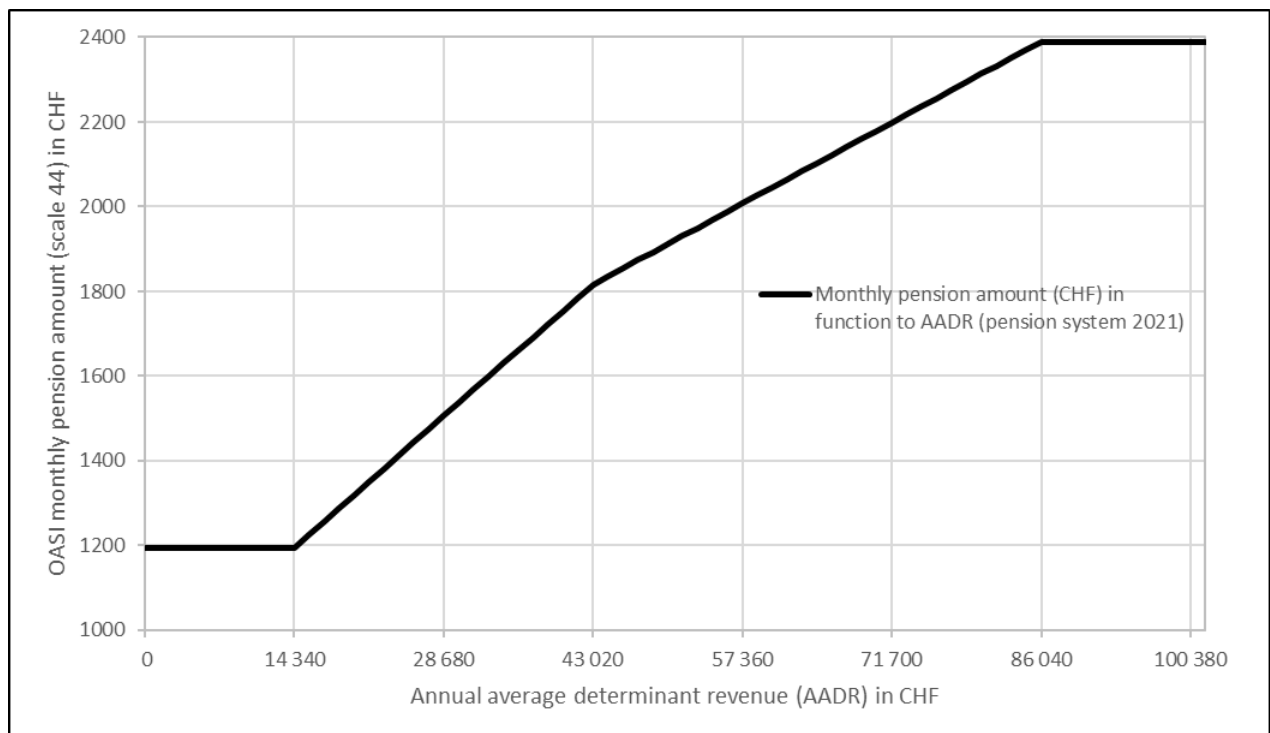


Figure A.6: OASI monthly pension amount in function to AADR for a minimal pension amount of 1195 CHF in effect in the year 2021

A.2 Descriptive Statistics

A.2.1 Cluster 1

1_aadr

	n	missing	distinct	Info	Mean	Gmd	.05	.10	.25	.50	.75	.90	.95
	754445	0	569	1	63953	29583	28440	35550	46926	58302	73944	93852	110916
lowest :	0	1422	2844	4266	5688	highest:	4034214	4322880	4991220	5188878	8708328		

1_age_retire

	n	missing	distinct	Info	Mean	Gmd
754445		0	9	0.733	64.17	0.6803

lowest : 62 63 64 65 66, highest: 66 67 68 69 70

Value	62	63	64	65	66	67	68	69	70
Frequency	26575	43257	473485	204475	2811	1291	683	1113	755
Proportion	0.035	0.057	0.628	0.271	0.004	0.002	0.001	0.001	0.001

1_age

	n	missing	distinct	Info	Mean	Gmd	.05	.10	.25	.50	.75	.90	.95
754445		0	38	0.999	76.93	9.825	65	66	70	76	83	89	92

lowest : 62 63 64 65 66, highest: 95 96 97 98 99

1_benef_type

	n	missing	distinct	Info	Mean	Gmd
754445		0	1	0	1	0

Value	1
Frequency	754445
Proportion	1

1_marital_stat

	n	missing	distinct	Info	Mean	Gmd
754445		0	3	0.843	2.61	1.451

Value	1	2	4
Frequency	261651	131727	361067
Proportion	0.347	0.175	0.479

1_monthly_pension

	n	missing	distinct	Info	Mean	Gmd	.05	.10	.25	.50	.75	.90	.95
754445		0	2745	0.983	2057	368.3	1316	1580	1891	2155	2370	2370	2370

lowest : 279 302 303 310 318, highest: 3428 3460 3464 3618 3635

1_nat

	n	missing	distinct	Info	Sum	Mean	Gmd
754445		0	2	0.267	74389	0.0986	0.1778

1_resid

	n	missing	distinct	Info	Sum	Mean	Gmd
754445		0	2	0.191	51463	0.06821	0.1271

1_scale

	n	missing	distinct	Info	Mean	Gmd	.05	.10	.25	.50	.75	.90	.95
	754445	0	33	0.474	42.44	2.789	33	38	44	44	44	44	44

lowest : 12 13 14 15 16, highest: 40 41 42 43 44

1_sex

	n	missing	distinct	Info	Sum	Mean	Gmd
	754445	0	2	0.638	523153	0.6934	0.4252

A.2.2 Cluster 2**2_aadr**

	n	missing	distinct	Info	Mean	Gmd	.05	.10	.25	.50	.75	.90	.95
	992117	0	741	1	69099	31616	34128	41238	49770	61146	78210	100962	122292

lowest : 0 1422 2844 4266 5688, highest: 7043166 7392978 7546554 7626186 11970396

2_age_retire

	n	missing	distinct	Info	Mean	Gmd
	992117	0	9	0.818	64.39	0.8001

lowest : 62 63 64 65 66, highest: 66 67 68 69 70

Value	62	63	64	65	66	67	68	69	70
Frequency	36262	61822	396197	486671	5025	2173	1161	1169	1637
Proportion	0.037	0.062	0.399	0.491	0.005	0.002	0.001	0.001	0.002

2_age

	n	missing	distinct	Info	Mean	Gmd	.05	.10	.25	.50	.75	.90	.95
	992117	0	38	0.998	73.67	7.558	65	66	68	73	78	83	86

lowest : 62 63 64 65 66, highest: 95 96 97 98 99

2_benef_type

	n	missing	distinct	Info	Mean	Gmd
	992117	0	1	0	1	0

Value	1
Frequency	992117
Proportion	1

2_marital_stat

	n	missing	distinct	Info	Mean	Gmd
992117		0	1	0	3	0

Value 3
 Frequency 992117
 Proportion 1

2_monthly_pension

	n	missing	distinct	Info	Mean	Gmd	.05	.10	.25	.50	.75	.90	.95
992117		0	2485	0.997	1744	237.7	1289	1493	1685	1778	1799	1953	2237

lowest : 303 311 321 326 327, highest: 3525 3550 3658 3677 3678

2_nat

	n	missing	distinct	Info	Sum	Mean	Gmd
992117		0	2	0.304	113512	0.1144	0.2026

2_resid

	n	missing	distinct	Info	Sum	Mean	Gmd
992117		0	2	0.191	67781	0.06832	0.1273

2_scale

	n	missing	distinct	Info	Mean	Gmd	.05	.10	.25	.50	.75	.90	.95
992117		0	32	0.482	42.44	2.766	32	38	44	44	44	44	44

lowest : 13 14 15 16 17, highest: 40 41 42 43 44

2_sex

	n	missing	distinct	Info	Sum	Mean	Gmd
992117		0	2	0.745	455442	0.4591	0.4966

A.2.3 Cluster 3**3_aadr**

	n	missing	distinct	Info	Mean	Gmd	.05	.10	.25	.50	.75	.90	.95
138843		0	448	1	50898	41431	12798	14220	19908	41238	66834	89586	110916

lowest : 0 1422 2844 4266 5688, highest: 2077542 2562444 6633630 10640826 11636226

3_age_retire

	n	missing	distinct	Info	Mean	Gmd
138843		0	1	0	-99999	0

Value -99999
 Frequency 138843
 Proportion 1

3_age

	n	missing	distinct	Info	Mean	Gmd	.05	.10	.25	.50	.75	.90	.95
138843		0	100	0.999	68.84	17.43	18	49	65	74	80	83	85

lowest : 0 1 2 3 4, highest: 95 96 97 98 99

3_benef_type

	n	missing	distinct	Info	Mean	Gmd
138843		0	7	0.531	2.814	1.323

lowest : 2 3 4 5 6, highest: 4 5 6 7 8

Value	2	3	4	5	6	7	8
Frequency	107627	5152	994	7	19675	5140	248
Proportion	0.775	0.037	0.007	0.000	0.142	0.037	0.002

3_marital_stat

	n	missing	distinct	Info	Mean	Gmd
138843		0	4	0.57	3.624	0.606

Value	1	2	3	4
Frequency	3162	11535	19620	104526
Proportion	0.023	0.083	0.141	0.753

3_monthly_pension

	n	missing	distinct	Info	Mean	Gmd	.05	.10	.25	.50	.75	.90	.95
138843		0	1330	1	324.3	372.5	16	22	49	165	474	914	1163

lowest : 7 8 9 10 11, highest: 1582 1594 1740 1795 1969

3_nat

	n	missing	distinct	Info	Sum	Mean	Gmd
138843		0	2	0.117	133215	0.9595	0.07778

3_resid

	n	missing	distinct	Info	Sum	Mean	Gmd
138843		0	2	0.09	134520	0.9689	0.06033

3_scale

	n	missing	distinct	Info	Mean	Gmd	.05	.10	.25	.50	.75	.90	.95
138843		0	38	0.994	9.936	9.84	1	1	2	7	15	25	30


lowest : 0 1 2 3 4, highest: 33 34 35 36 37

3_sex

	n	missing	distinct	Info	Sum	Mean	Gmd
138843		0	2	0.122	132944	0.9575	0.08136

A.2.4 Cluster 4


4_aadr



n	missing	distinct	Info	Mean	Gmd	.05	.10	.25	.50	.75	.90	.95
692197	0	760	1	43720	32781	11376	14220	21330	35550	54036	75366	93852

lowest : 0 1422 2844 4266 5688, highest: 5825934 6504228 10640826 14787378 17647020

4_age_retire




n	missing	distinct	Info	Mean	Gmd
692197	0	9	0.805	64.39	0.7098

lowest : 62 63 64 65 66, highest: 66 67 68 69 70

Value	62	63	64	65	66	67	68	69	70
Frequency	17025	38579	299035	335411	1003	362	185	288	309
Proportion	0.025	0.056	0.432	0.485	0.001	0.001	0.000	0.000	0.000

4_age



n	missing	distinct	Info	Mean	Gmd	.05	.10	.25	.50	.75	.90	.95
692197	0	38	0.998	75.35	8.024	65	66	70	75	80	85	88


lowest : 62 63 64 65 66, highest: 95 96 97 98 99

4_benef_type

n	missing	distinct	Info	Mean	Gmd
692197	0	1	0	1	0

Value 1
Frequency 692197
Proportion 1


4_marital_stat



n	missing	distinct	Info	Mean	Gmd
692197	0	4	0.701	2.94	0.724

Value 1 2 3 4
Frequency 62922 42842 459310 127123
Proportion 0.091 0.062 0.664 0.184

4_monthly_pension



n	missing	distinct	Info	Mean	Gmd	.05	.10	.25	.50	.75	.90	.95
692197	0	1974	1	415.9	443.6	30	54	87	238	608	1107	1382

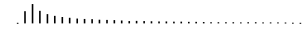
lowest : 23 24 25 26 27, highest: 2399 2467 2475 2543 2558

4_nat

n	missing	distinct	Info	Sum	Mean	Gmd
692197	0	2	0.203	641753	0.9271	0.1351

4_resid

	n	missing	distinct	Info	Sum	Mean	Gmd
4_scale	692197	0	2	0.132	660245	0.9538	0.08806

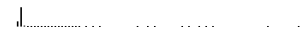


	n	missing	distinct	Info	Mean	Gmd	.05	.10	.25	.50	.75	.90	.95
4_scale	692197	0	39	0.994	9.827	9.643	1	2	3	6	14	25	30

lowest : 0 1 2 3 4, highest: 34 35 36 37 38

4_sex

	n	missing	distinct	Info	Sum	Mean	Gmd
4_sex	692197	0	2	0.746	320938	0.4637	0.4974

A.2.5 Cluster 5**5_aadr**

	n	missing	distinct	Info	Mean	Gmd	.05	.10	.25	.50	.75	.90	.95
5_aadr	111005	0	482	1	81124	44370	29862	38394	55458	72522	92430	120870	149310

lowest : 0 1422 2844 4266 5688, highest: 5496030 5655294 5682312 7306236 8190720

5_age_retire

	n	missing	distinct	Info	Mean	Gmd
5_age_retire	111005	0	1	0	-99999	0

Value -99999
Frequency 111005
Proportion 1

5_age

	n	missing	distinct	Info	Mean	Gmd	.05	.10	.25	.50	.75	.90	.95
5_age	111005	0	100	0.999	42.48	27.14	10	13	18	51	62	74	80

lowest : 0 1 2 3 4, highest: 95 96 97 98 99

5_benef_type

	n	missing	distinct	Info	Mean	Gmd
5_benef_type	111005	0	7	0.812	3.368	1.923

lowest : 2 3 4 5 6, highest: 4 5 6 7 8

	Value	2	3	4	5	6	7	8
Frequency	62127	18679	5931	25	1310	21609	1324	
Proportion	0.560	0.168	0.053	0.000	0.012	0.195	0.012	

5_marital_stat

	n	missing	distinct	Info	Mean	Gmd
111005		0	4	0.807	2.907	1.138

Value	1	2	3	4
Frequency	8301	47545	1313	53846
Proportion	0.075	0.428	0.012	0.485

5_monthly_pension

	n	missing	distinct	Info	Mean	Gmd	.05	.10	.25	.50	.75	.90	.95
111005		0	1713	0.997	1237	582.3	462	582	791	1255	1744	1896	1896

lowest : 129 140 143 144 146, highest: 2654 2672 2675 2804 2816

5_nat

	n	missing	distinct	Info	Sum	Mean	Gmd
111005		0	2	0.557	27340	0.2463	0.3713

5_resid

	n	missing	distinct	Info	Sum	Mean	Gmd
111005		0	2	0.529	25391	0.2287	0.3528

5_scale

	n	missing	distinct	Info	Mean	Gmd	.05	.10	.25	.50	.75	.90	.95
111005		0	34	0.717	40.65	5.33	26	31	40	44	44	44	44

lowest : 11 12 13 14 15, highest: 40 41 42 43 44

5_sex

	n	missing	distinct	Info	Sum	Mean	Gmd
111005		0	2	0.527	85798	0.7729	0.351

A.3 Histograms

A.3.1 Female old-age pensioners

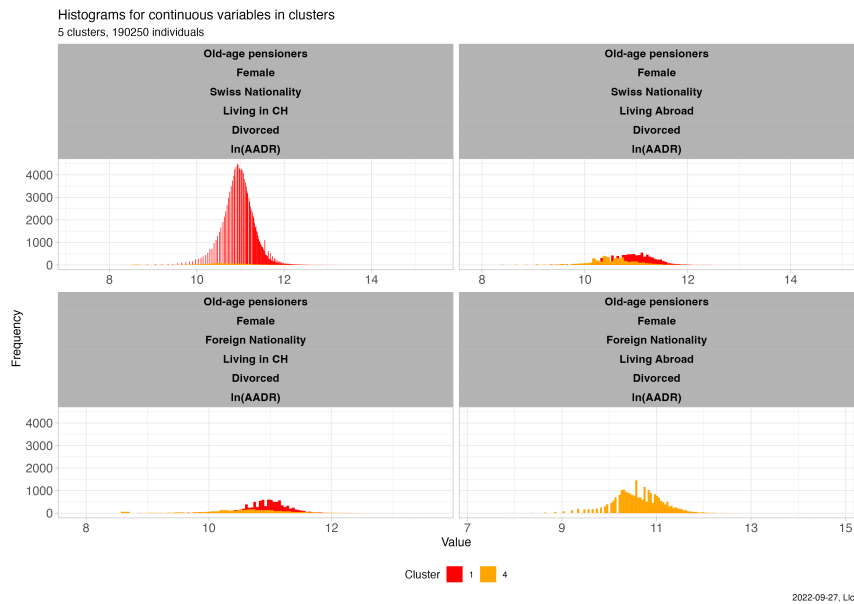


Figure A.7: Histogram for divorced female old-age pensioners

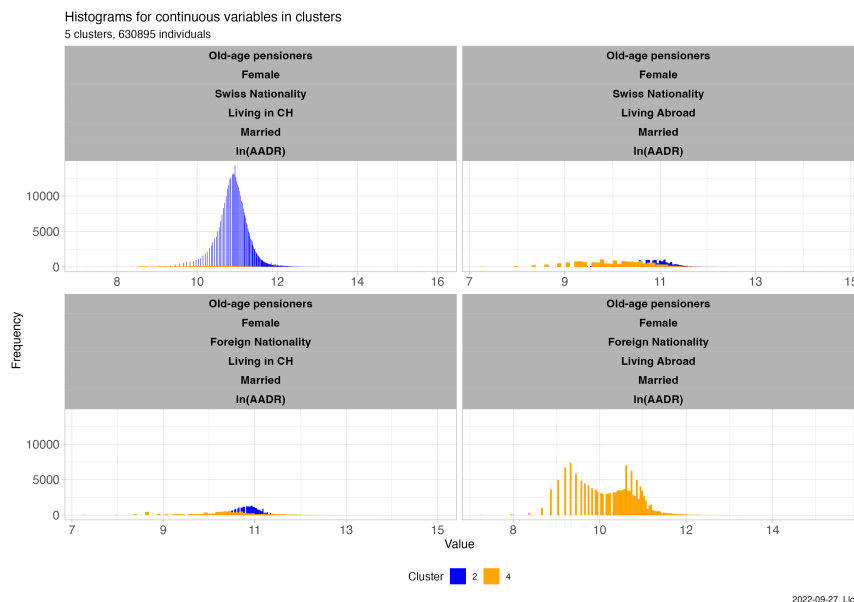


Figure A.8: Histogram for married female old-age pensioners

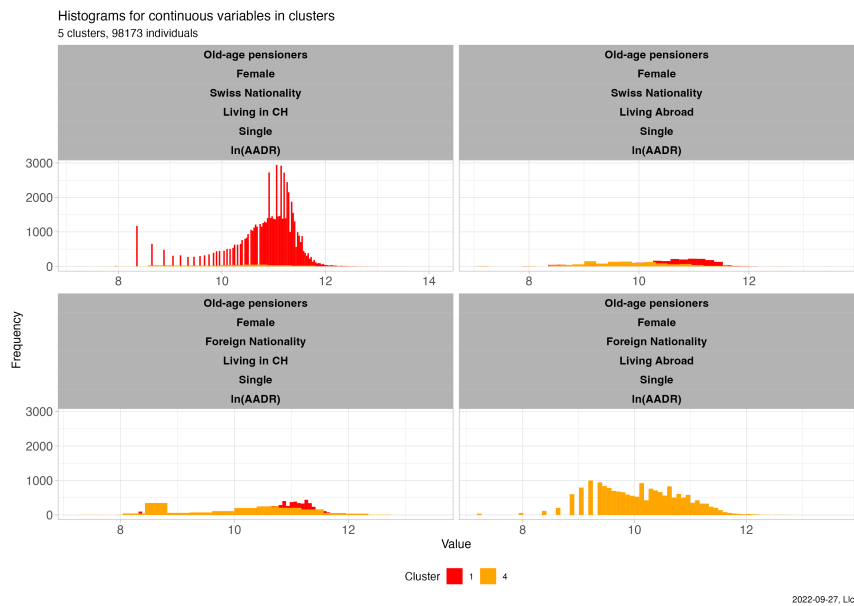


Figure A.9: Histogram for single female old-age pensioners

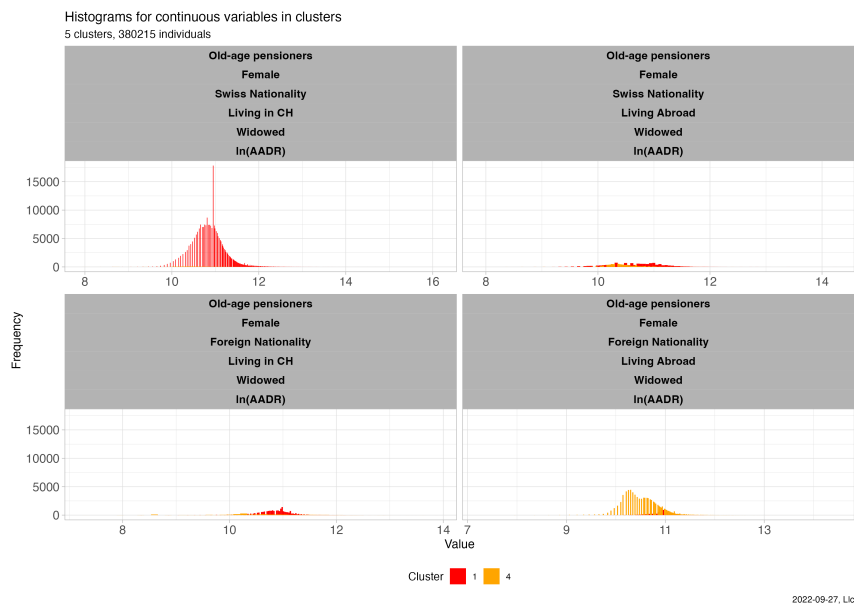


Figure A.10: Histogram for widowed female old-age pensioners

A.3.2 Other types of OASI female pensioners

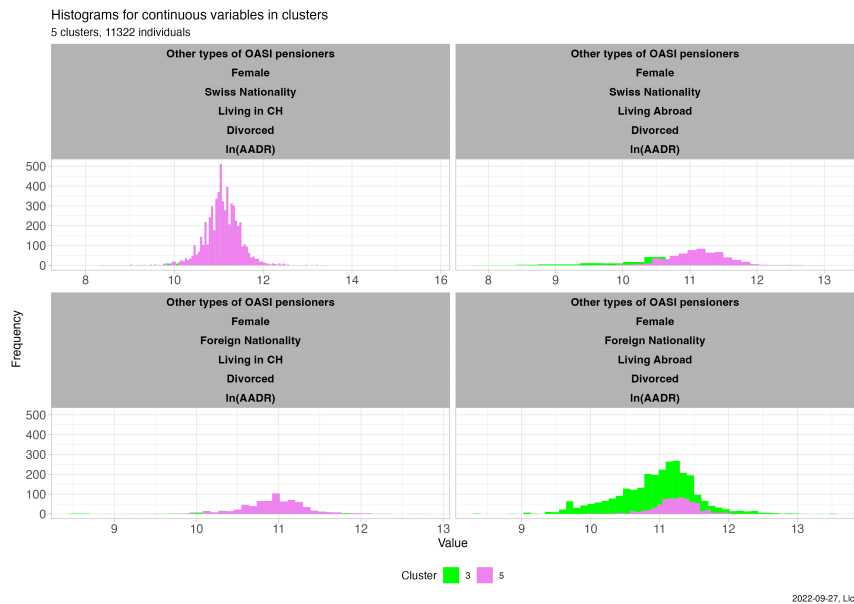


Figure A.11: Histogram for divorced female pensioners getting another type of OASI pension

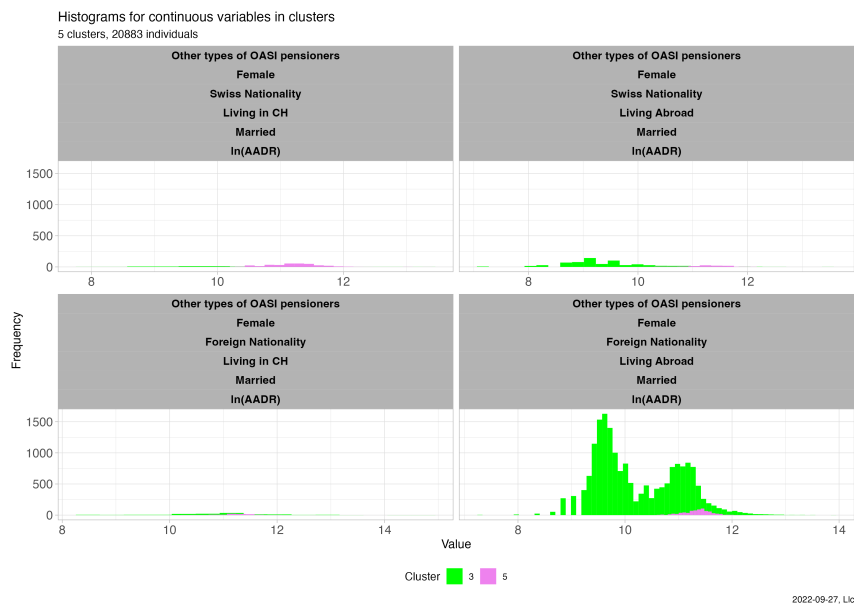


Figure A.12: Histogram for married female pensioners getting another type of OASI pension

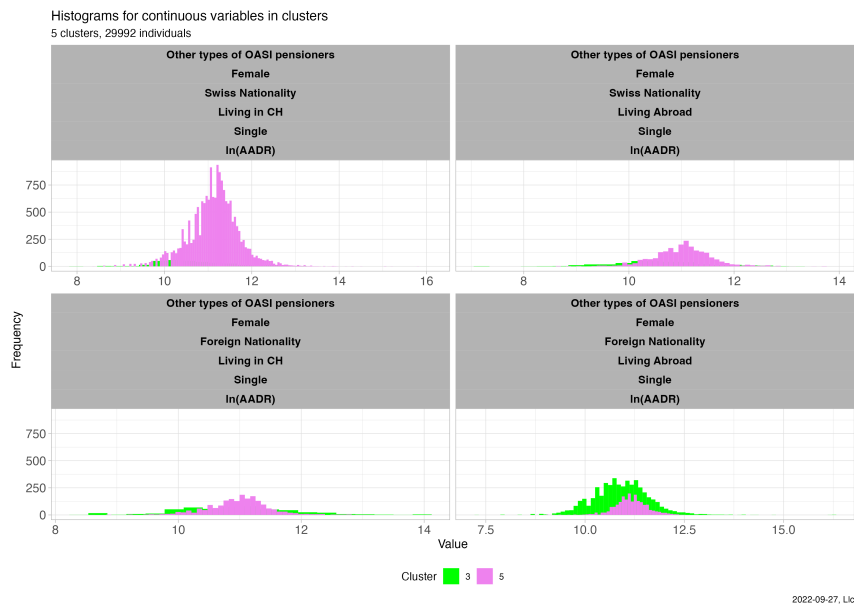


Figure A.13: Histogram for single female pensioners getting another type of OASI pension

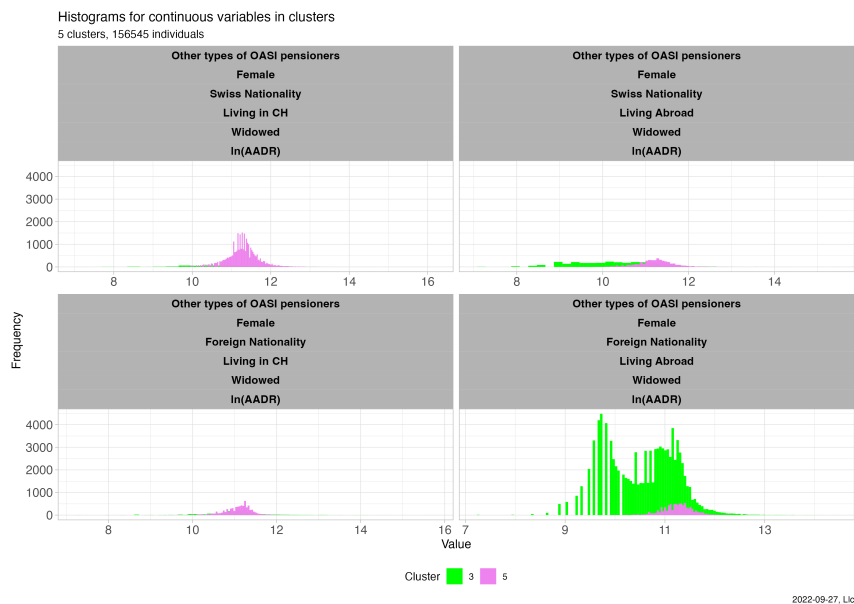


Figure A.14: Histogram for widowed female pensioners getting another type of OASI pension

A.3.3 Male old-age pensioners

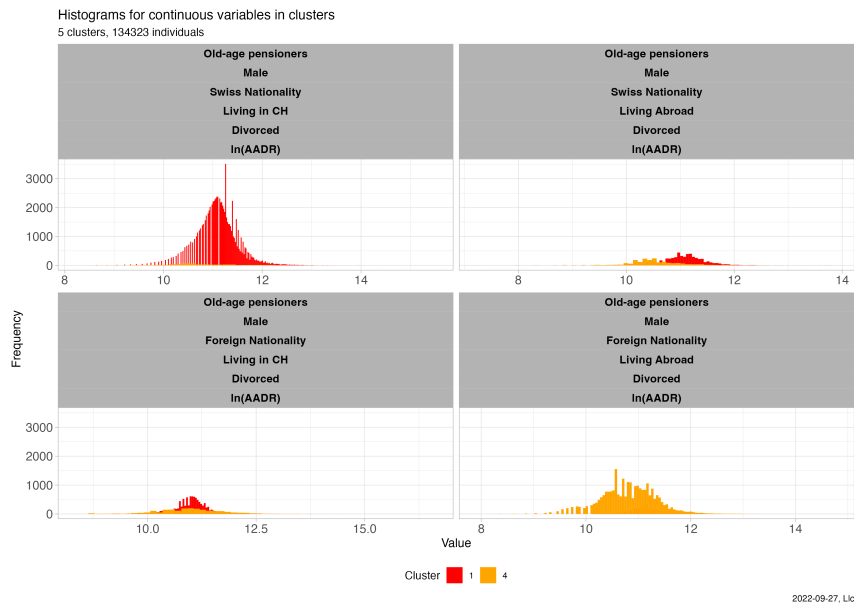


Figure A.15: Histogram for divorced male old-age pensioners

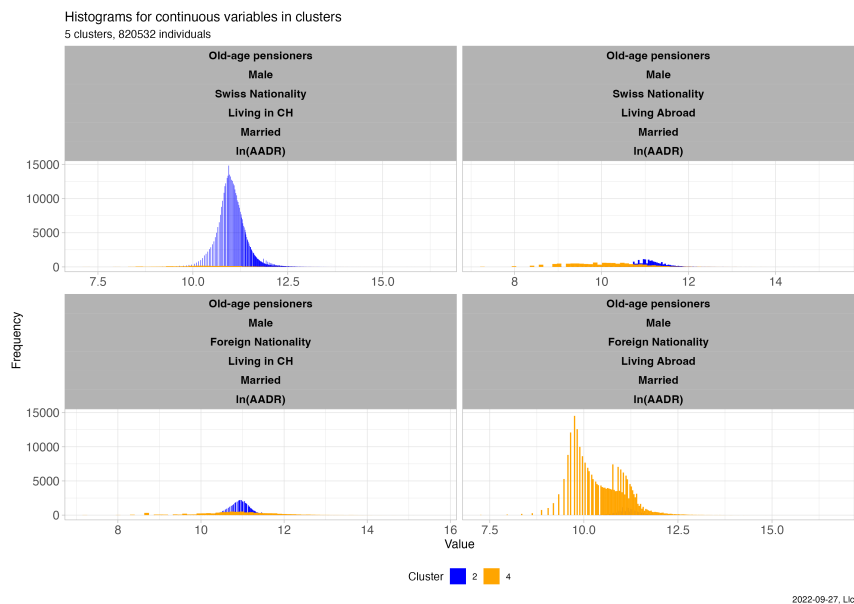


Figure A.16: Histogram for married male old-age pensioners

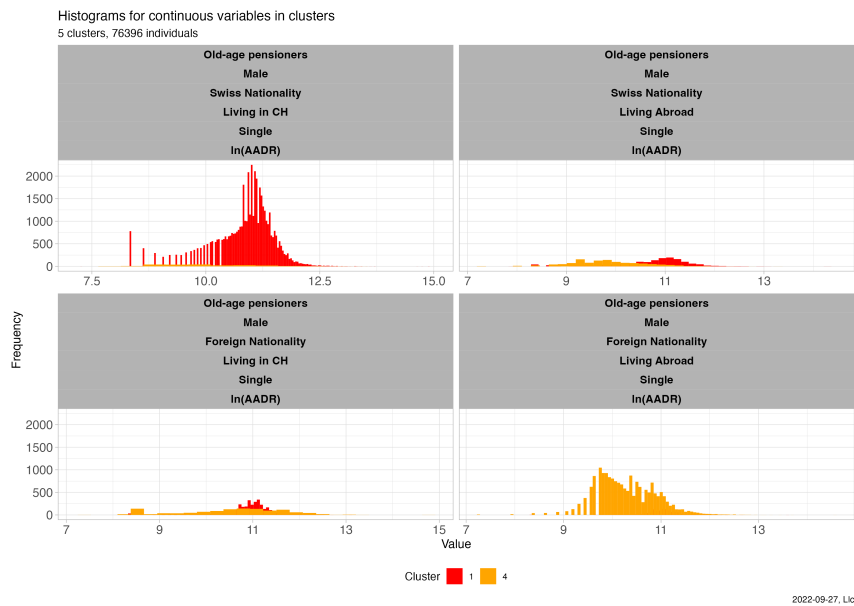


Figure A.17: Histogram for single male old-age pensioners

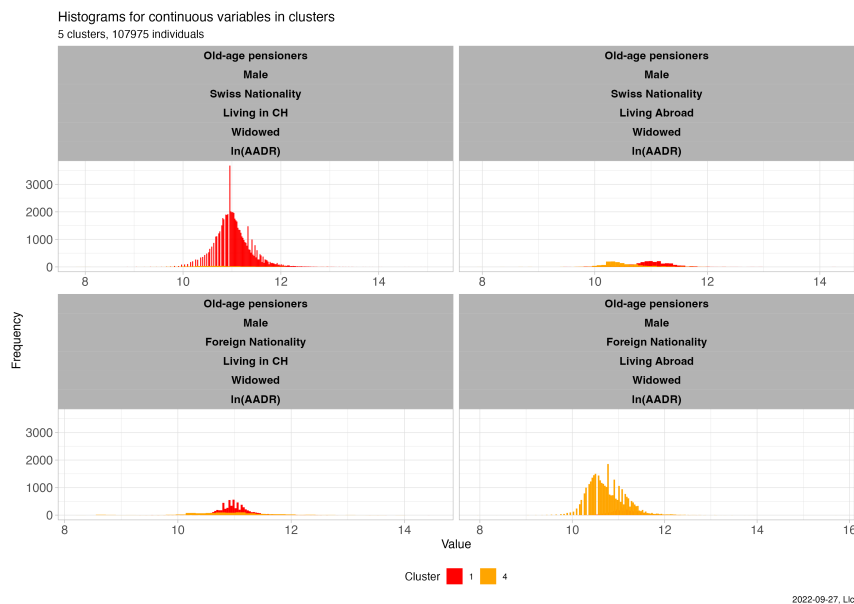


Figure A.18: Histogram for widowed male old-age pensioners

A.3.4 Other types of OASI male pensioners

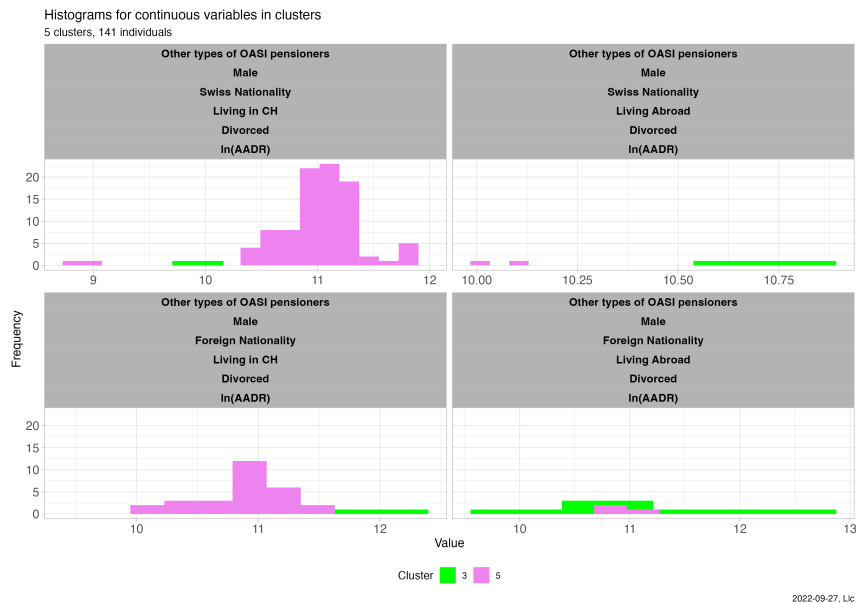


Figure A.19: Histogram for divorced male pensioners getting another type of OASI pension

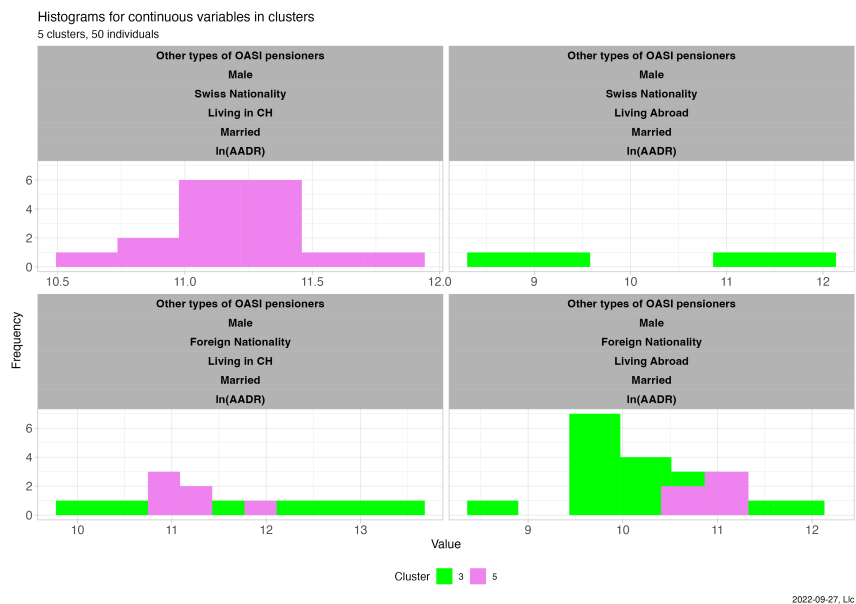


Figure A.20: Histogram for married male pensioners getting another type of OASI pension

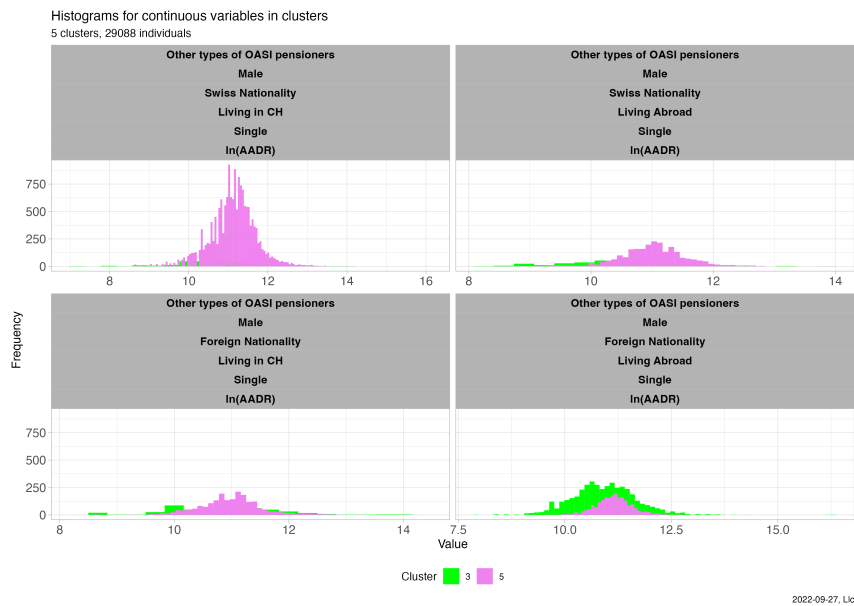


Figure A.21: Histogram for single male pensioners getting another type of OASI pension

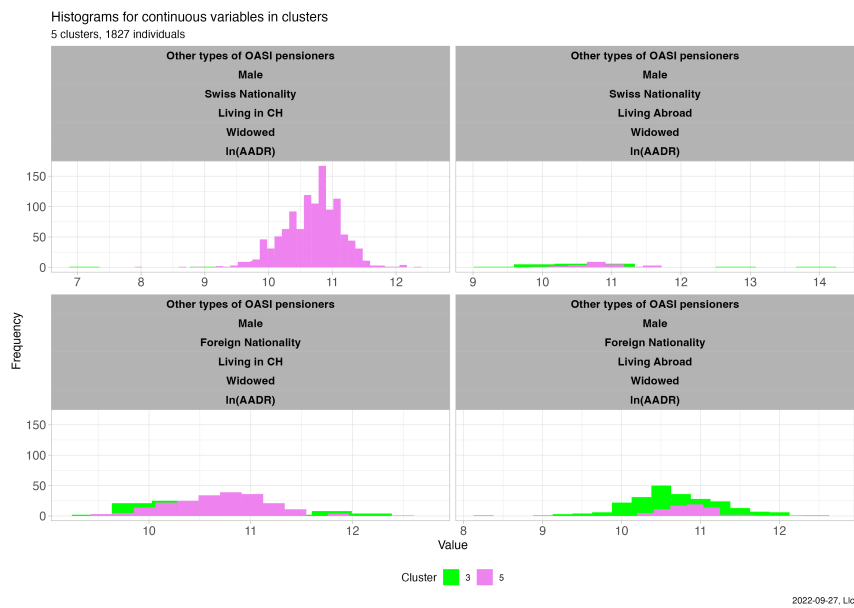


Figure A.22: Histogram for widowed male pensioners getting another type of OASI pension

A.4 Clusters Contingency Table



Table A.1: Clusters Contingency Table

Sex	Nationality	Residence	Pensioner Type	Marital Status		Clusters				
						1	2	3	4	5
Male	Swiss	Switzerland	Old-age pensioners	Divorced		84328	0	0	289	0
				Single		48712	0	0	299	0
				Married		0	442405	0	1942	0
				Widowed		61380	0	0	177	0
			Other types of OASI pensioners	Divorced		0	0	2	0	94
				Single		0	0	291	0	16916
				Married		0	0	0	0	17
				Widowed		0	0	24	0	1140
	Foreign Country		Old-age pensioners	Divorced		5707	0	0	2082	0
				Single		1592	0	0	1193	0
				Married		0	21652	0	11490	0
				Widowed		2420	0	0	1511	0
			Other types of OASI pensioners	Divorced		0	0	2	0	2
				Single		0	0	339	0	2406
				Married		0	0	2	0	0
				Widowed		0	0	20	0	25
	Total		Old-age pensioners	Divorced		90035	0	0	2371	0
				Single		50304	0	0	1492	0
				Married		0	464057	0	13432	0
				Widowed		63800	0	0	1688	0
			Other types of OASI pensioners	Divorced		0	0	4	0	96
				Single		0	0	630	0	19322
				Married		0	0	2	0	17
				Widowed		0	0	44	0	1165
	Foreign	Switzerland	Old-age pensioners	Divorced		10263	0	0	2093	0
				Single		3744	0	0	1211	0
				Married		0	53853	0	8575	0
				Widowed		6591	0	0	817	0
			Other types of OASI pensioners	Divorced		0	0	4	0	28
				Single		0	0	503	0	2079
				Married		0	0	3	0	6
				Widowed		0	0	96	0	215

Foreign Country	Old-age pensioners	Divorced	2833	0	0	0	26728	0
		Single	1153	0	0	0	18492	0
		Married	0	18765	0	0	261850	0
		Widowed	2569	0	0	0	32510	0
		Divorced	0	0	6	0	0	3
		Single	0	0	4356	0	0	2198
		Married	0	0	17	0	0	5
		Widowed	0	0	234	0	0	73
		Divorced	13096	0	0	0	28821	0
		Single	4897	0	0	0	19703	0
Total	Old-age pensioners	Married	0	72618	0	0	270425	0
		Widowed	9160	0	0	0	33327	0
		Divorced	0	0	10	0	0	31
		Single	0	0	4859	0	0	4277
		Married	0	0	20	0	0	11
		Widowed	0	0	330	0	0	288
		Divorced	94591	0	0	0	2382	0
		Single	52456	0	0	0	1510	0
		Married	0	496258	0	0	10517	0
		Widowed	67971	0	0	0	994	0
Foreign Country	Old-age pensioners	Divorced	0	0	6	0	0	122
		Single	0	0	794	0	0	18995
		Married	0	0	3	0	0	23
		Widowed	0	0	120	0	0	1355
		Divorced	8540	0	0	0	28810	0
		Single	2745	0	0	0	19685	0
		Married	0	40417	0	0	273340	0
		Widowed	4989	0	0	0	34021	0
		Divorced	0	0	8	0	0	5
		Single	0	0	4695	0	0	4604
Total	Old-age pensioners	Married	0	0	19	0	0	5
		Widowed	0	0	254	0	0	98
		Divorced	103131	0	0	0	31192	0
		Single	55201	0	0	0	21195	0
		Married	0	536675	0	0	283857	0
Foreign Country	Other types of OASI pensioners	Divorced	0	0	0	0	0	0
		Single	0	0	0	0	0	0
		Married	0	0	0	0	0	0
		Widowed	0	0	0	0	0	0
		Divorced	0	0	0	0	0	0
		Single	0	0	0	0	0	0
		Married	0	0	0	0	0	0
		Widowed	0	0	0	0	0	0
		Divorced	0	0	0	0	0	0
		Single	0	0	0	0	0	0
Total	Other types of OASI pensioners	Married	0	0	0	0	0	0
		Widowed	0	0	0	0	0	0
		Divorced	0	0	0	0	0	0
		Single	0	0	0	0	0	0
		Married	0	0	0	0	0	0
		Widowed	0	0	0	0	0	0
		Divorced	0	0	0	0	0	0
		Single	0	0	0	0	0	0
		Married	0	0	0	0	0	0
		Widowed	0	0	0	0	0	0

Female	Swiss	Switzerland	Other types of OASI pensioners	Widowed	72960	0	0	35015	0
				Divorced	0	0	14	0	127
				Single	0	0	5489	0	23599
				Married	0	0	22	0	28
				Widowed	0	0	374	0	1453
			Old-age pensioners	Divorced	137148	0	0	543	0
				Single	66501	0	0	354	0
				Married	0	394904	0	1524	0
				Widowed	248243	0	0	481	0
				Other types of OASI pensioners	Divorced	0	0	61	0
Foreign Country			Single	0	0	348	0	17693	
			Married	0	0	35	0	340	
			Widowed	0	0	456	0	31171	
			Divorced	9383	0	0	4316	0	
			Single	2201	0	0	1176	0	
			Married	0	19644	0	16208	0	
			Widowed	12441	0	0	6859	0	
			Other types of OASI pensioners	Divorced	0	0	179	0	597
			Single	0	0	328	0	2318	
			Married	0	0	681	0	106	
Total			Widowed	0	0	2860	0	4697	
			Divorced	146531	0	0	4859	0	
			Single	68702	0	0	1530	0	
			Married	0	414548	0	17732	0	
			Widowed	260684	0	0	7340	0	
			Other types of OASI pensioners	Divorced	0	0	240	0	6740
			Single	0	0	676	0	20011	
			Married	0	0	716	0	446	
			Widowed	0	0	3316	0	35868	
			Foreign	Switzerland	Old-age pensioners	Divorced	9614	0	0
Single	5993	0				0	1571	0	
Married	0	33174				0	8722	0	
Widowed	20465	0				0	1841	0	
Other types of OASI pensioners	Divorced	0				0	90	0	698
Single	0	0			549	0	1935		

Foreign Country	Old-age pensioners	Married	0	0	129	0	89
		Widowed	0	0	1732	0	7050
		Divorced	2375	0	0	25358	0
		Single	1831	0	0	18546	0
		Married	0	7720	0	148999	0
		Widowed	6958	0	0	82927	0
		Divorced	0	0	2818	0	736
		Single	0	0	4821	0	2000
		Married	0	0	18753	0	750
		Widowed	0	0	99104	0	9475
Total	Old-age pensioners	Divorced	11989	0	0	26871	0
		Single	7824	0	0	20117	0
		Married	0	40894	0	157721	0
		Widowed	27423	0	0	84768	0
		Divorced	0	0	2908	0	1434
		Single	0	0	5370	0	3935
		Married	0	0	18882	0	839
		Widowed	0	0	100836	0	16525
		Divorced	146762	0	0	2056	0
		Single	72494	0	0	1925	0
Switzerland	Old-age pensioners	Married	0	428078	0	10246	0
		Widowed	268708	0	0	2322	0
		Divorced	0	0	151	0	6841
		Single	0	0	897	0	19628
		Married	0	0	164	0	429
		Widowed	0	0	2188	0	38221
		Divorced	11758	0	0	29674	0
		Single	4032	0	0	19722	0
		Married	0	27364	0	165207	0
		Widowed	19399	0	0	89786	0
Foreign Country	Old-age pensioners	Divorced	0	0	2997	0	1333
		Single	0	0	5149	0	4318
		Married	0	0	19434	0	856
		Widowed	0	0	101964	0	14172
		Divorced	158520	0	0	31730	0
Total	Old-age pensioners	Divorced	158520	0	0	31730	0
		Single	0	0	0	0	0
		Married	0	0	0	0	0
		Widowed	0	0	0	0	0
		Divorced	0	0	0	0	0
		Single	0	0	0	0	0
		Married	0	0	0	0	0
		Widowed	0	0	0	0	0
		Divorced	0	0	0	0	0
		Single	0	0	0	0	0

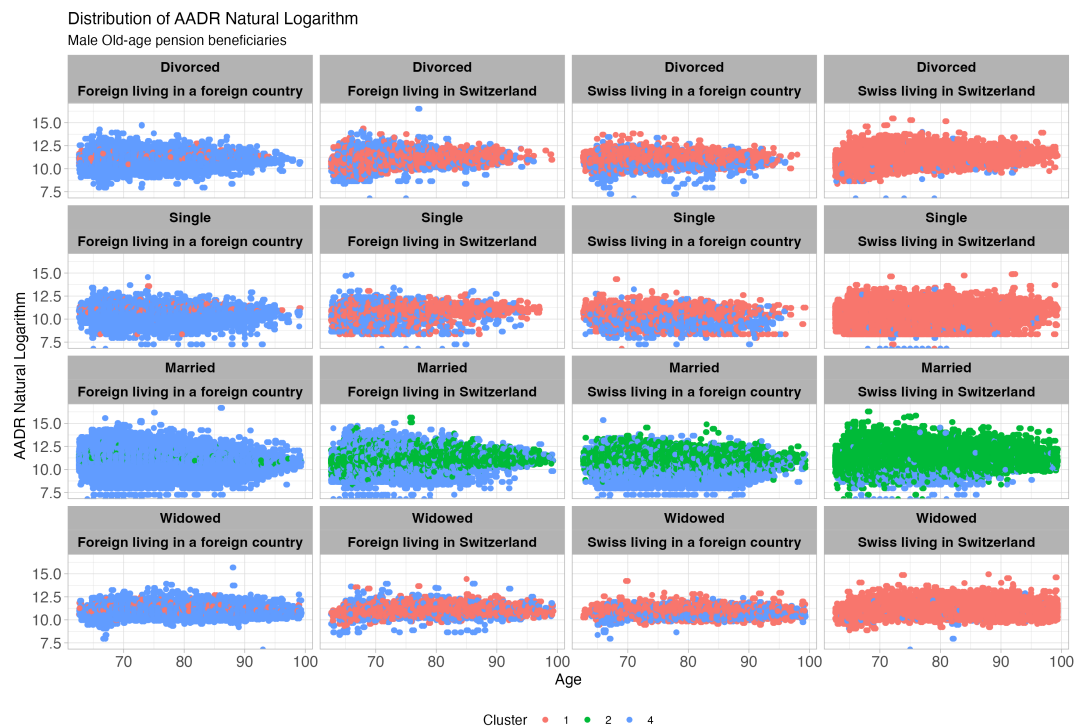
Other types of OASI pensioners	Single	76526	0	0	21647	0
	Married	0	455442	0	175453	0
	Widowed	288107	0	0	92108	0
	Divorced	0	0	3148	0	8174
	Single	0	0	6046	0	23946
	Married	0	0	19598	0	1285
	Widowed	0	0	104152	0	52393

A.5 Clusters Scatterplots of the AADR



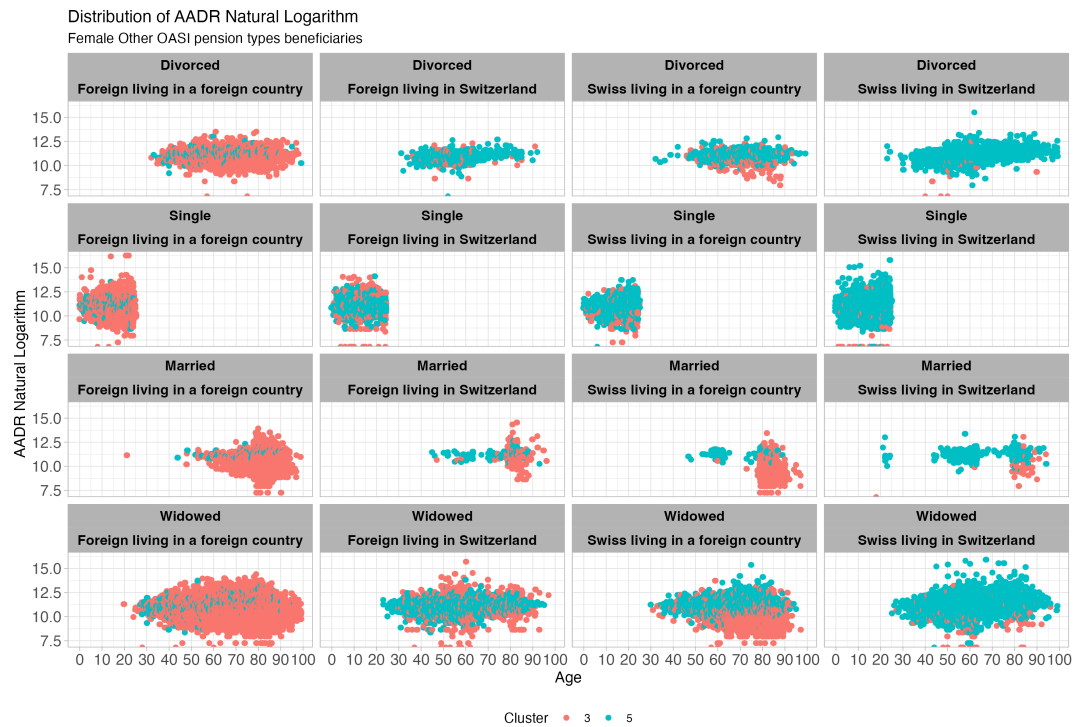


(a) Males getting a type of pension different from old-age: AADR distribution

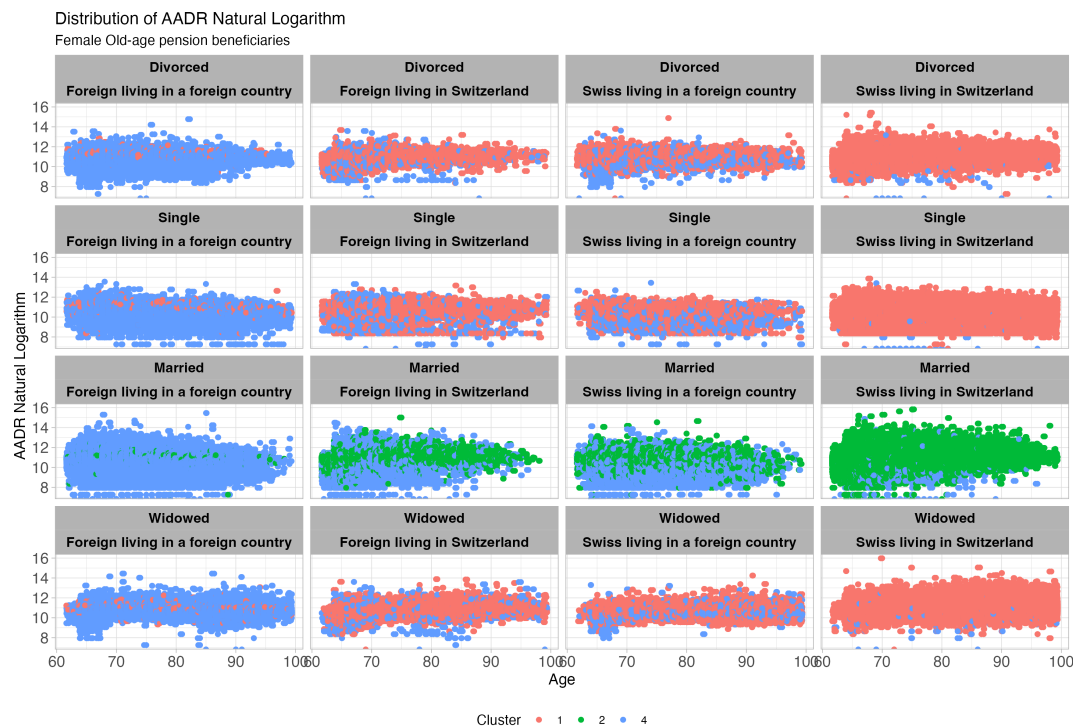


(b) Male Old-age insurance AADR distribution

Figure A.23: Male AADR distribution according to the age

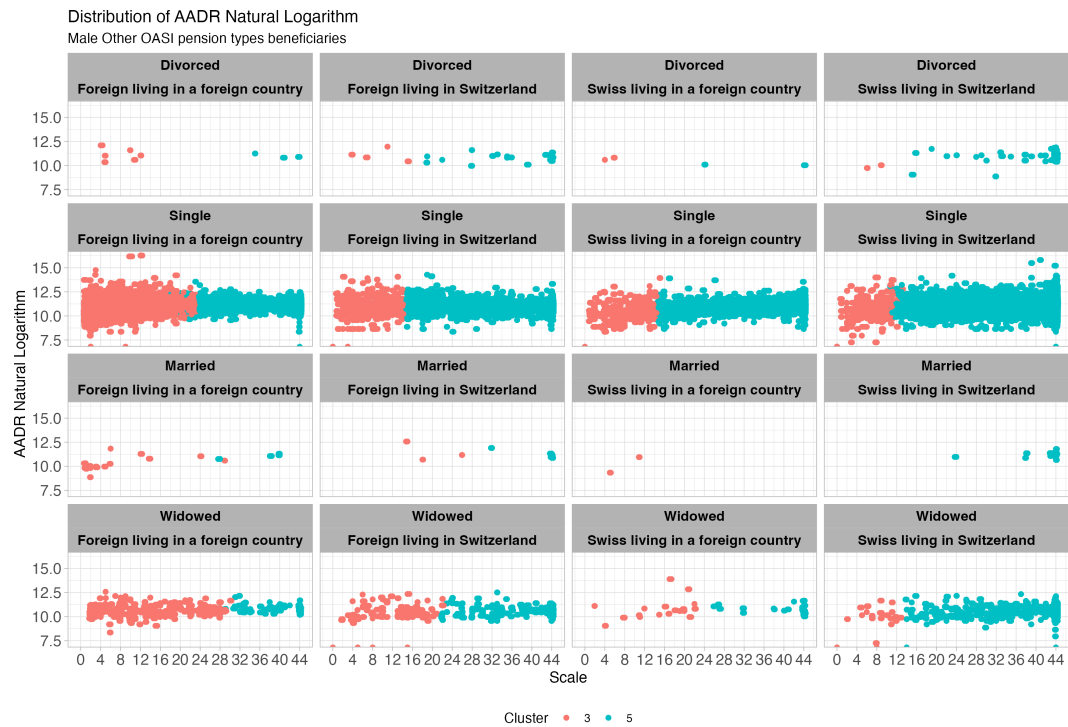


(a) Female getting a type of pension different from old-age: AADR distribution

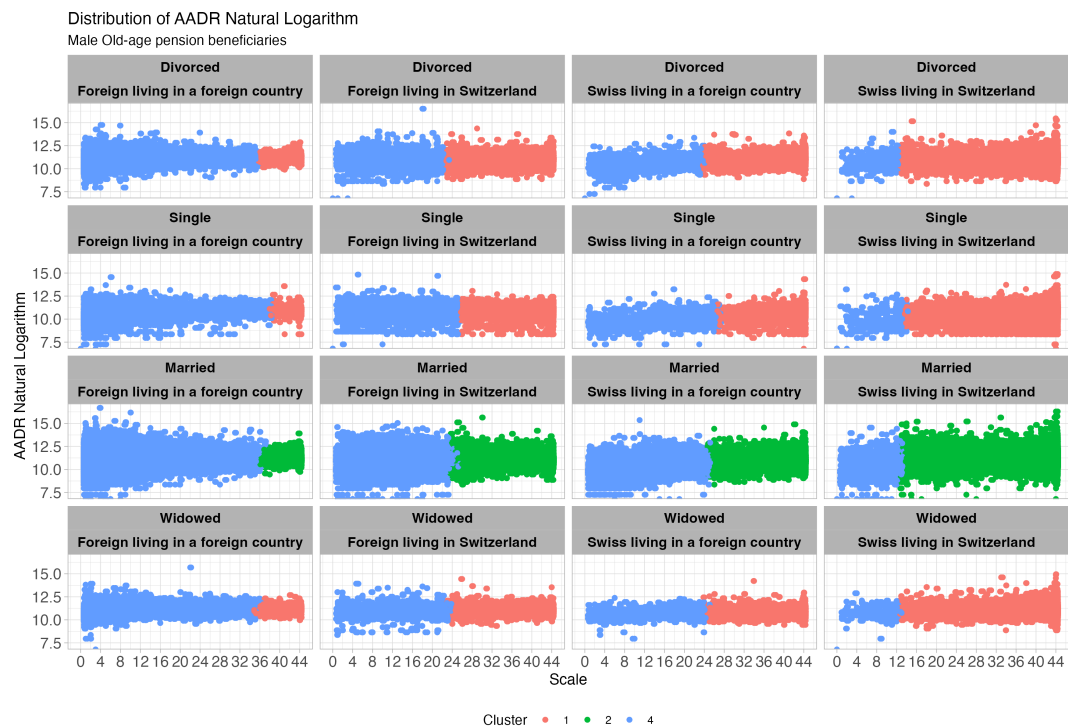


(b) Female Old-age insurance AADR distribution

Figure A.24: Female AADR distribution according to the age



(a) Males getting a type of pension different from old-age: AADR distribution

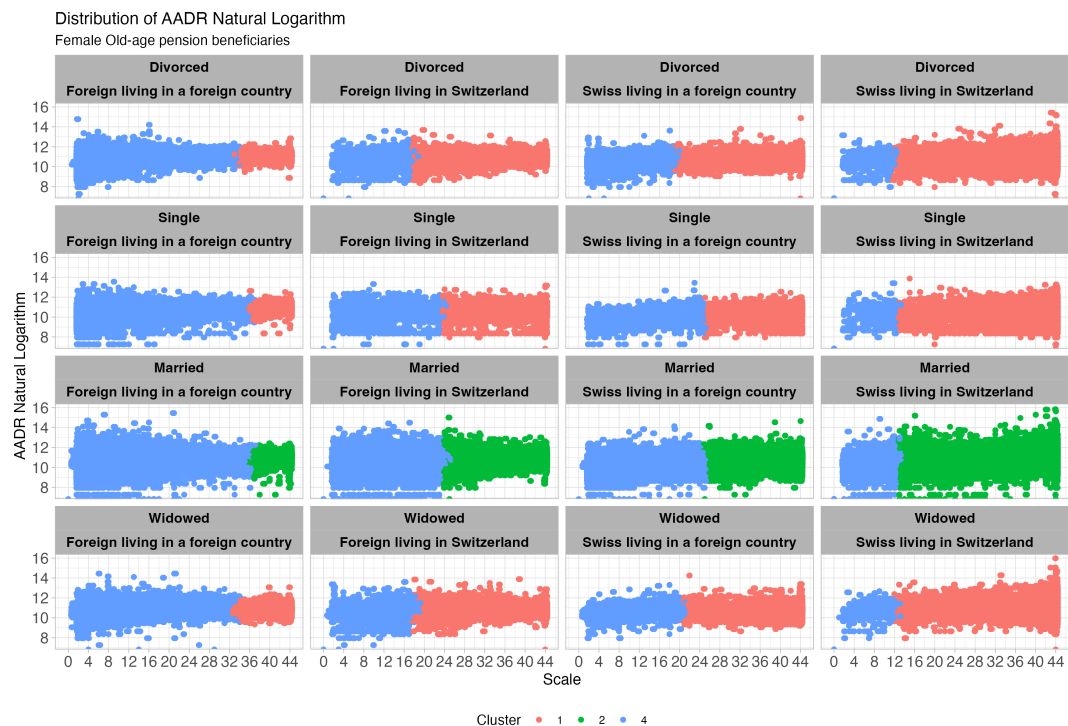


(b) Male Old-age insurance AADR distribution

Figure A.25: Male AADR distribution according to the scale

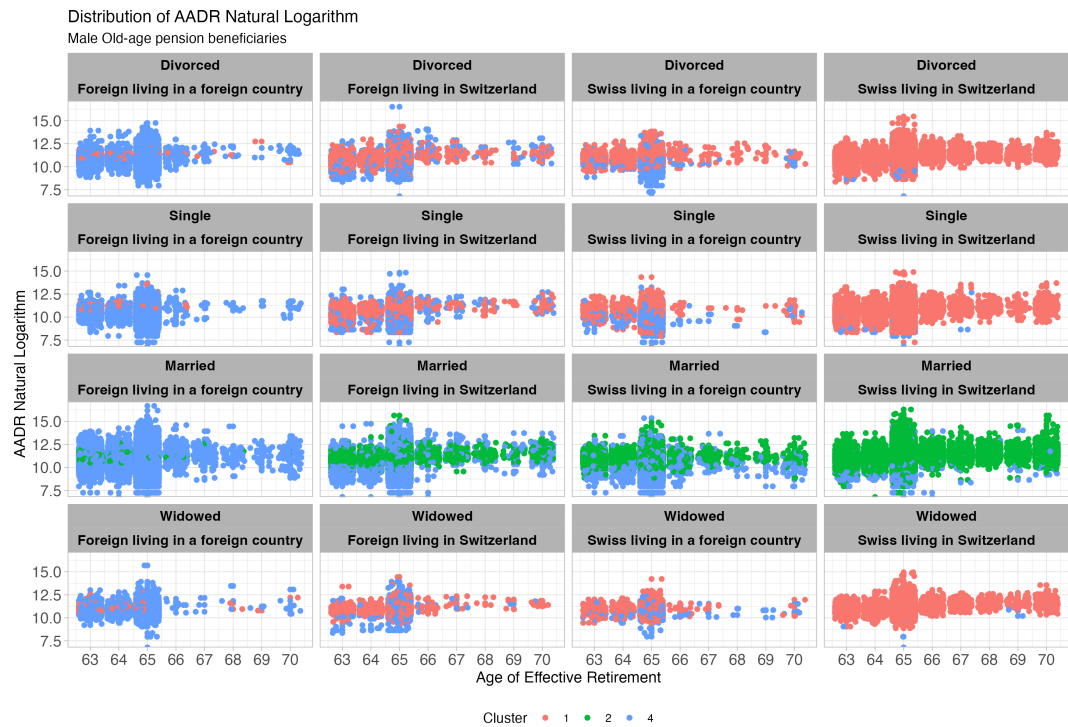


(a) Females getting a type of pension different from old-age: AADR distribution

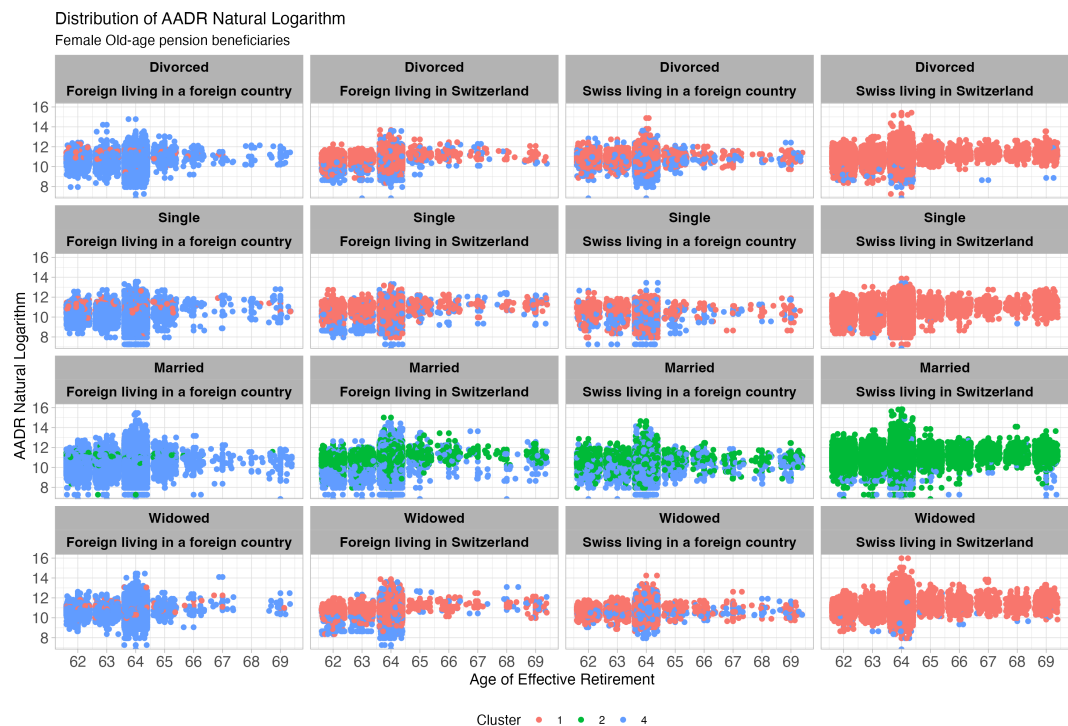


(b) Female Old-age insurance AADR distribution

Figure A.26: Female AADR distribution according to the scale



(a) Male Old-age insurance AADR distribution

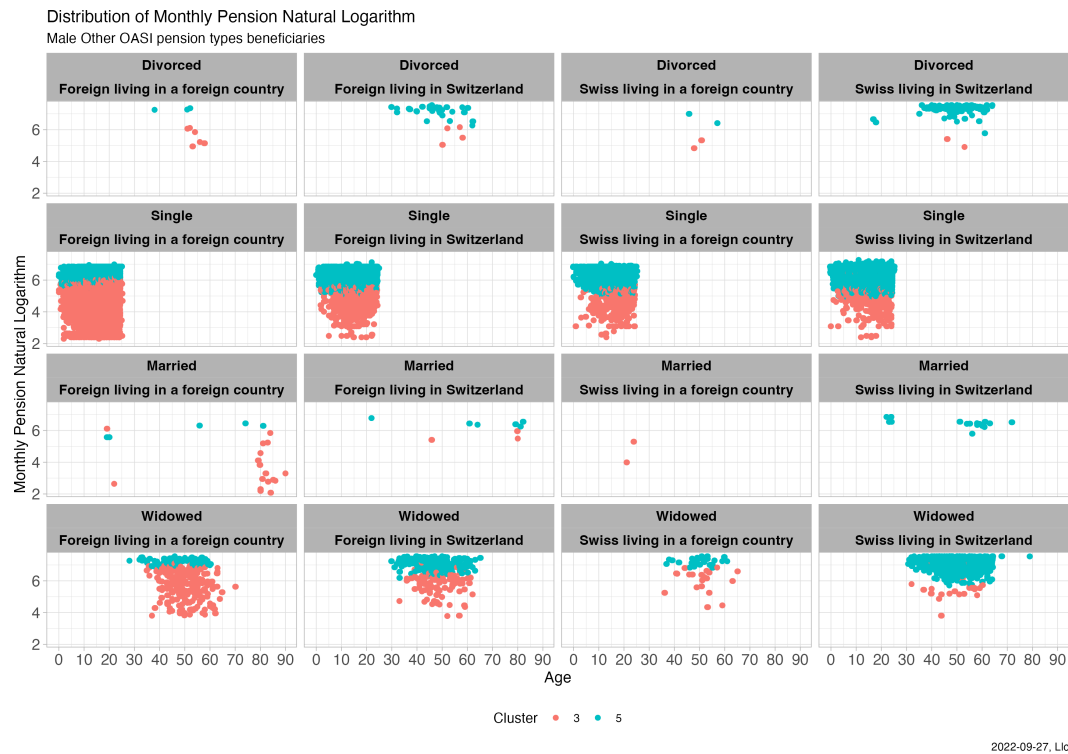


(b) Female Old-age insurance AADR distribution

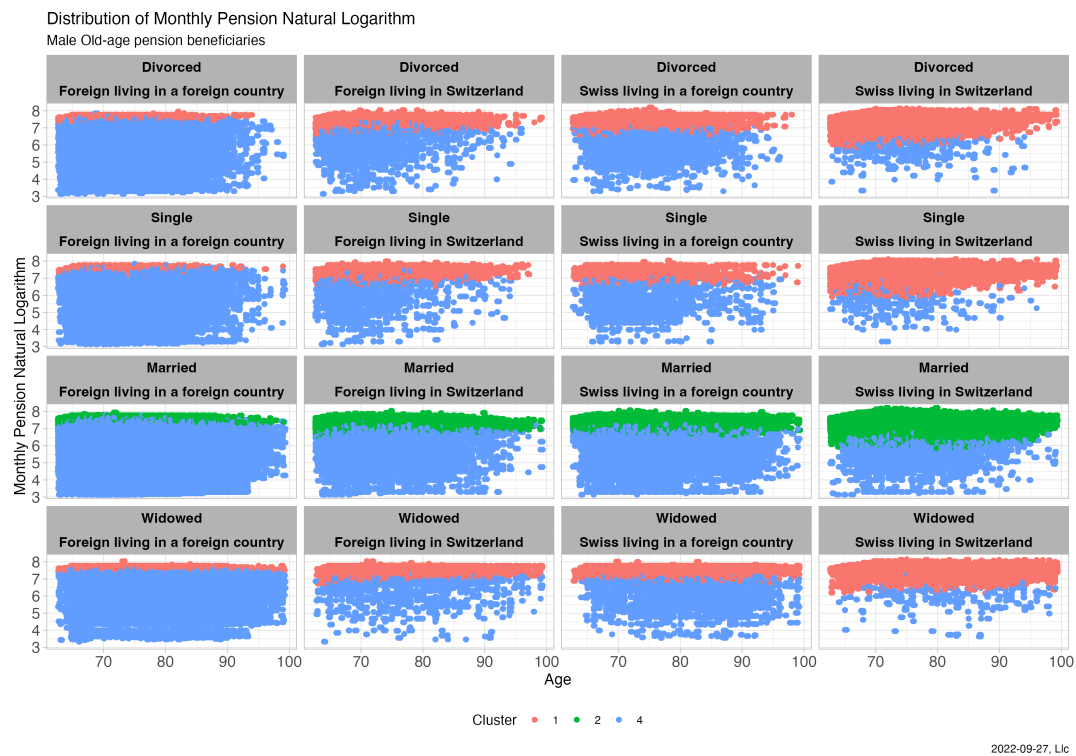
Figure A.27: Male and Female AADR distribution according to the age of retirement

A.6 Clusters Scatterplots of the monthly pension amount





(a) Males getting a type of pension different from old-age: monthly pension amount distribution

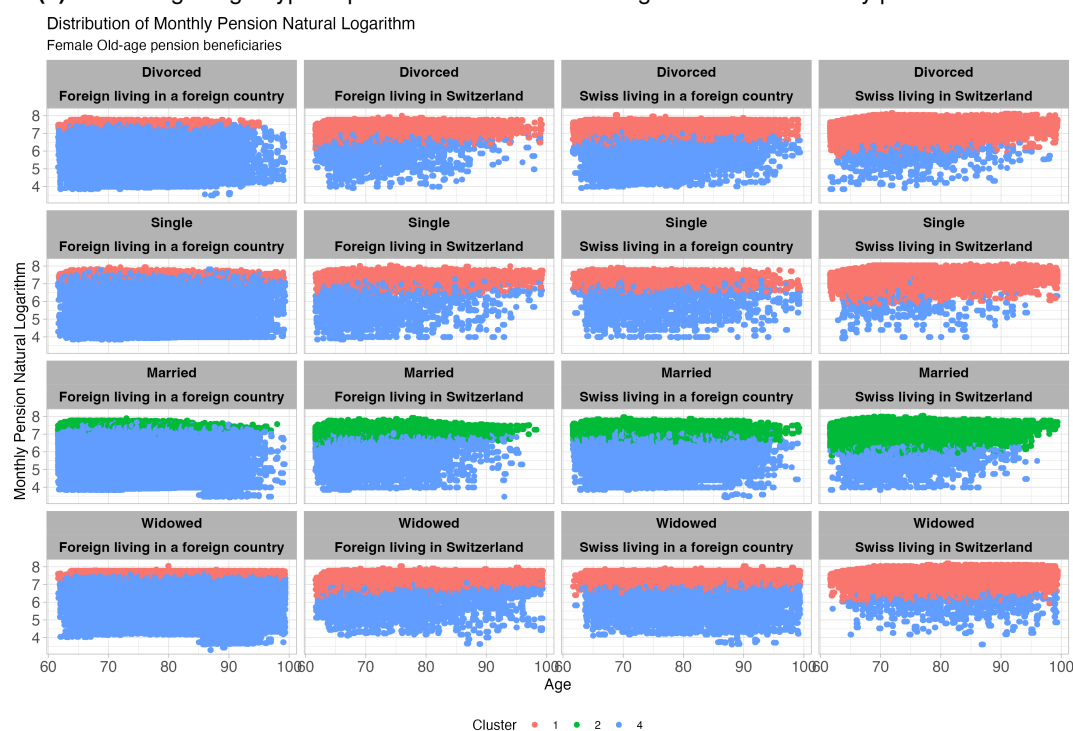


(b) Male Old-age insurance monthly pension amount distribution

Figure A.28: Male monthly pension amount distribution according to the age

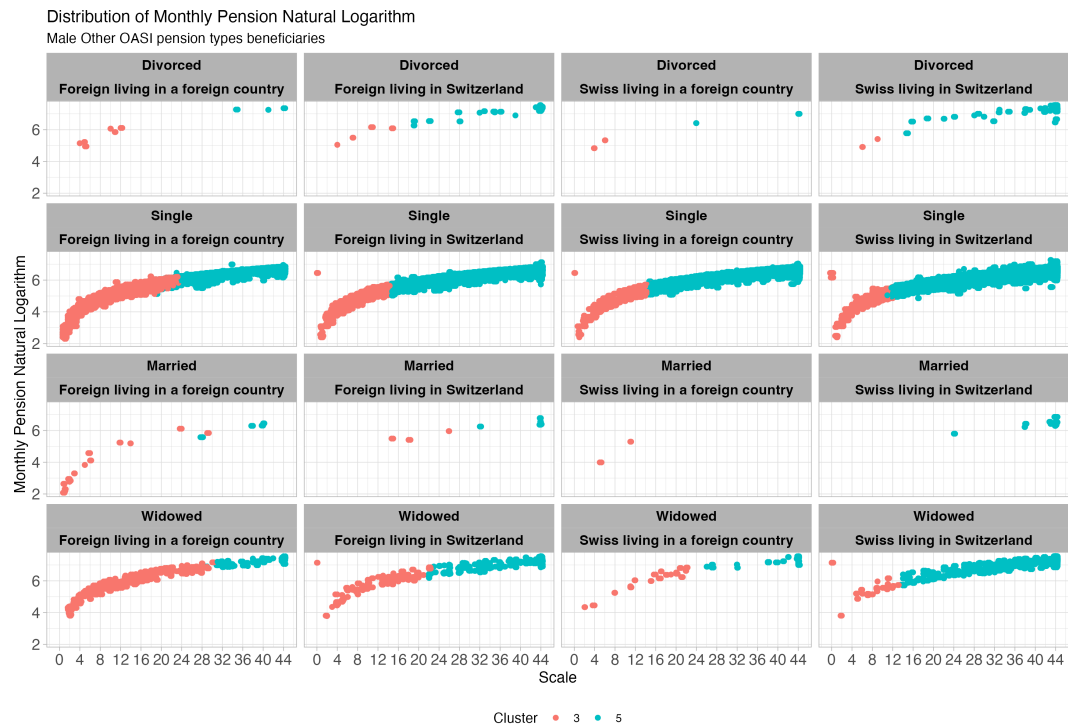


(a) Females getting a type of pension different from old-age: insurance monthly pension amount distribution



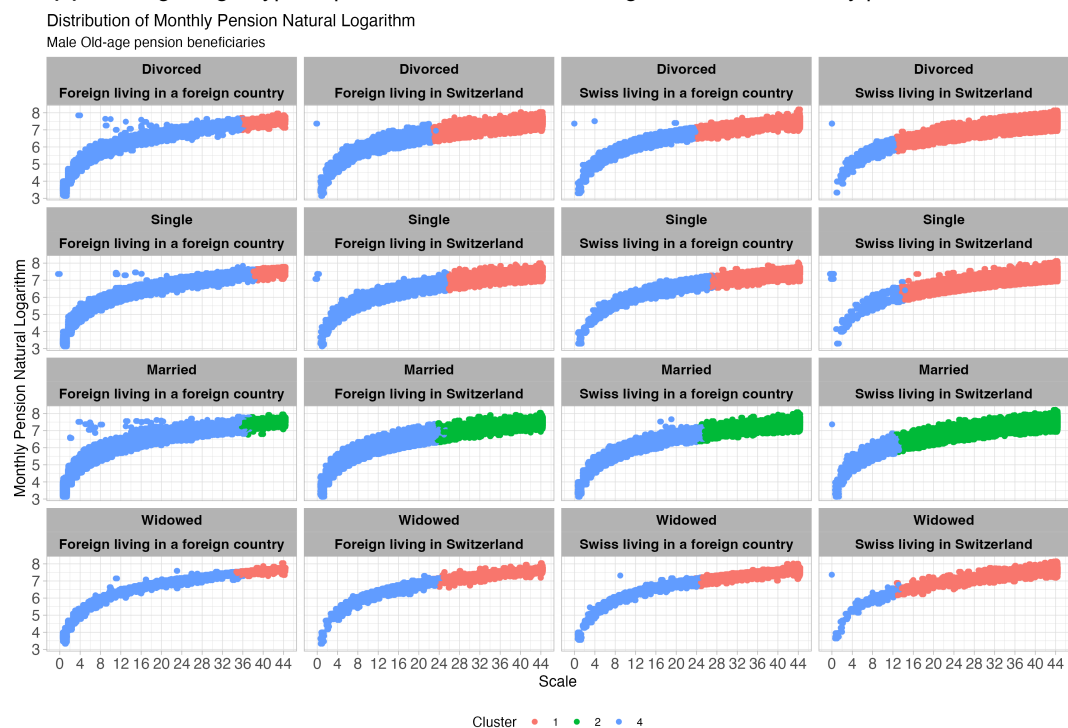
(b) Female Old-age insurance monthly pension amount distribution

Figure A.29: Female monthly pension amount distribution according to the age



2022-09-27, Lic

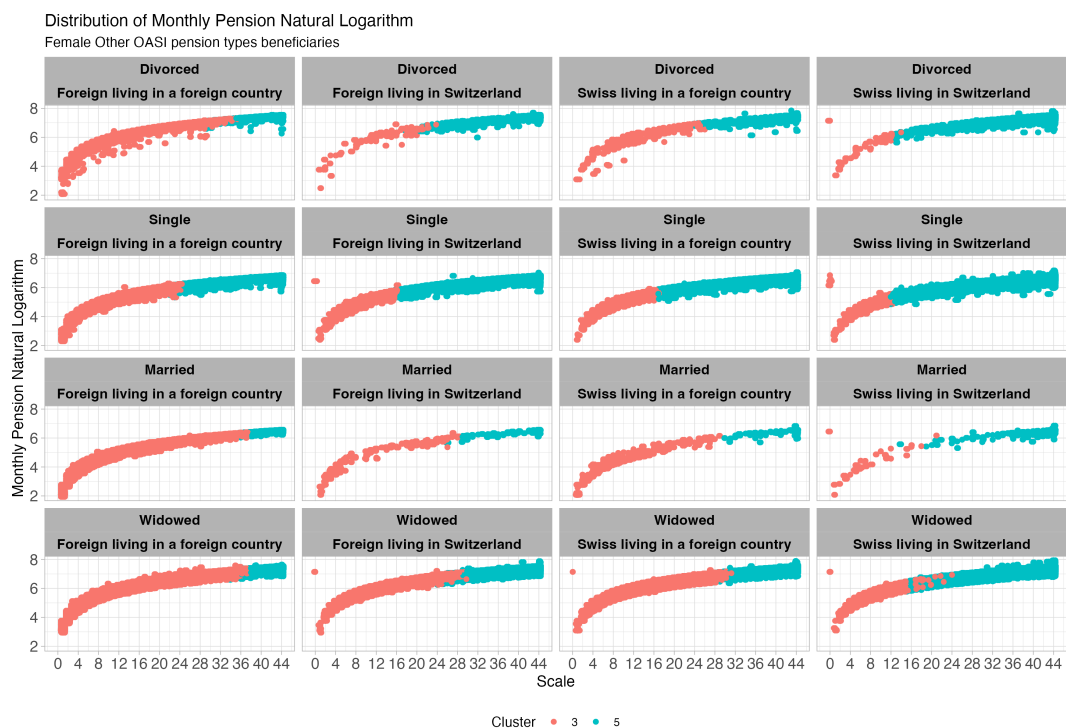
(a) Males getting a type of pension different from old-age: insurance monthly pension amount distribution



2022-09-27, Lic

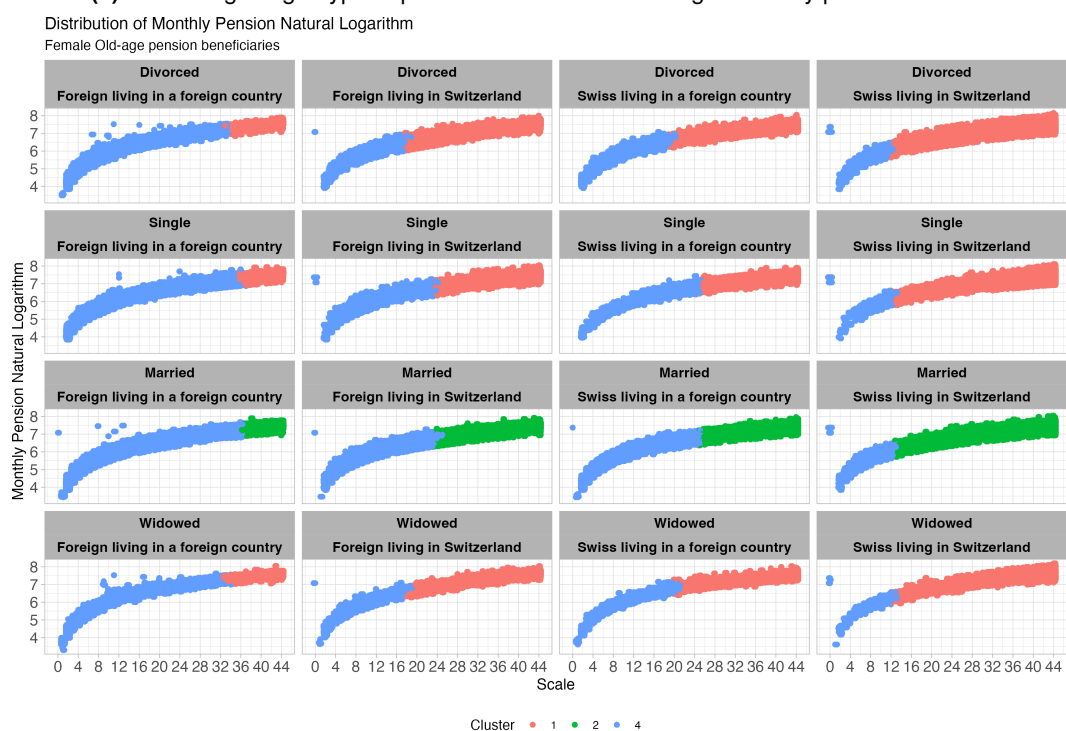
(b) Male Old-age insurance monthly pension amount distribution

Figure A.30: Male monthly pension amount distribution according to the scale



2022-09-27, Lic

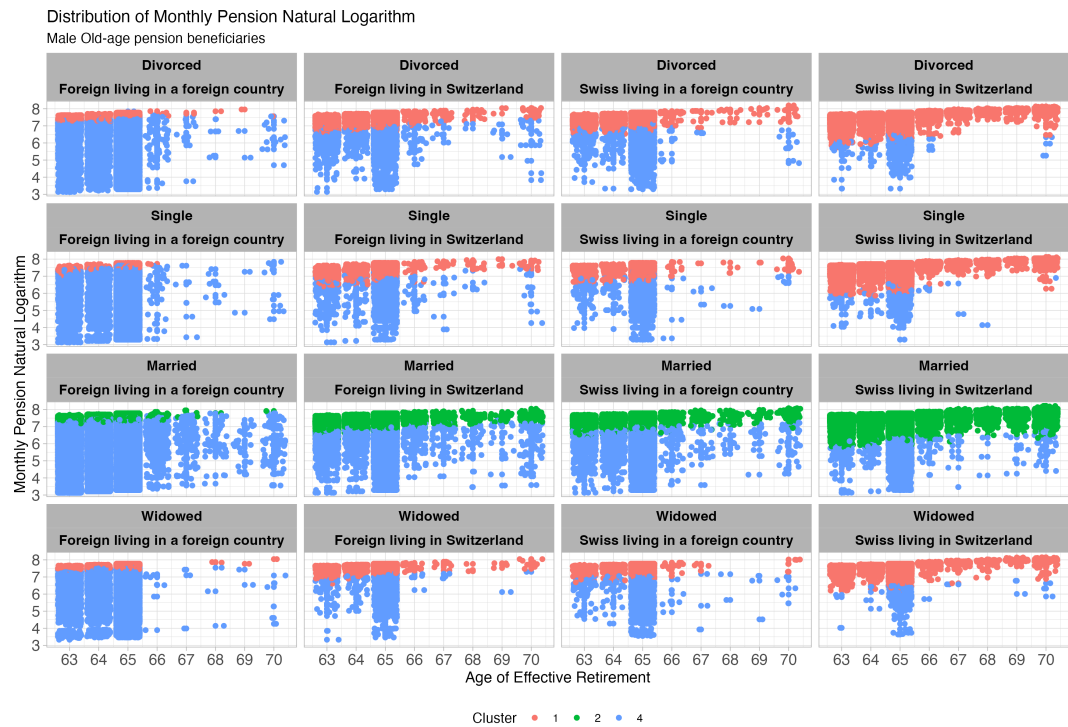
(a) Females getting a type of pension different from old-age: monthly pension amount distribution



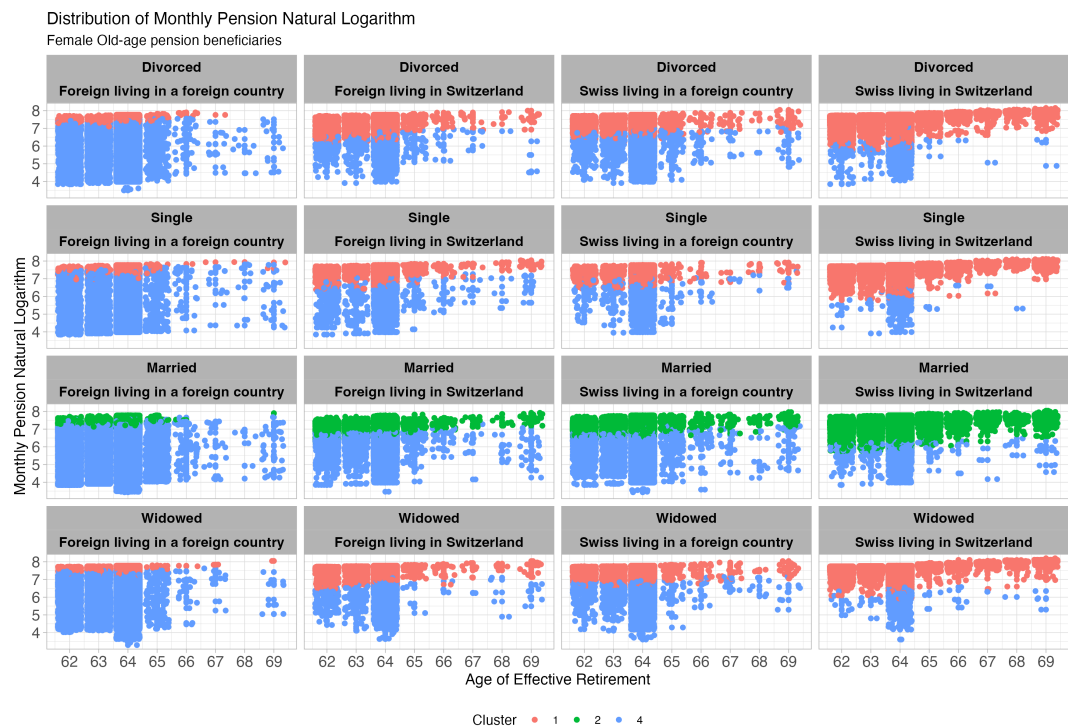
2022-09-27, Lic

(b) Female Old-age insurance monthly pension amount distribution

Figure A.31: Female monthly pension amount distribution according to the scale



(a) Male Old-age insurance monthly pension amount distribution



(b) Female Old-age insurance monthly pension amount distribution

Figure A.32: Male and Female monthly pension amount distribution according to the age of retirement

A.7 Clusters Summary Statistics

Variable	N	Mean	Std. Dev.	Min	Pctl. 25	Pctl. 75	Max
year	754445	2020	0	2020	2020	2020	2020
age	754445	76.927	8.601	62	70	83	99
sex	754445	0.693	0.461	0	0	1	1
nat	754445	0.099	0.298	0	0	0	1
resid	754445	0.068	0.252	0	0	0	1
aadr	754445	63953.25	38834.003	0	46926	73944	8708328
monthly_pension	754445	2056.64	352.59	279	1891	2370	3635
capping	754445	0	0	0	0	0	0
contrib_m_ind	754445	468.245	87.488	12	456	516	540
contrib_y_ageclass	754445	40.773	5.691	1	41	44	45
splitting	754445	0.643	0.479	0	0	1	1
bonus_m_edu	754445	72.428	70.755	0	0	126	504
bonus_m_assist	754445	0.151	3.255	0	0	0	264
benef_type	754445	1	0	1	1	1	1
marital_stat	754445	2.61	1.374	1	1	4	4
scale	754445	42.436	4.307	12	44	44	44
marital_stat1	754445	0.347	0.476	0	0	1	1
marital_stat2	754445	0.175	0.38	0	0	0	1
marital_stat3	754445	0	0	0	0	0	0
marital_stat4	754445	0.479	0.5	0	0	1	1
benef_type1	754445	1	0	1	1	1	1
benef_type2	754445	0	0	0	0	0	0
benef_type3	754445	0	0	0	0	0	0
benef_type4	754445	0	0	0	0	0	0
benef_type5	754445	0	0	0	0	0	0
benef_type6	754445	0	0	0	0	0	0
benef_type7	754445	0	0	0	0	0	0
benef_type8	754445	0	0	0	0	0	0
cluster_id	754445	1	0	1	1	1	1

Table A.2: Summary Statistics of the Clustered Pension Register, Cluster 1

Variable	N	Mean	Std. Dev.	Min	Pctl. 25	Pctl. 75	Max
year	992117	2020	0	2020	2020	2020	2020
age	992117	73.674	6.717	62	68	78	99
sex	992117	0.459	0.498	0	0	1	1
nat	992117	0.114	0.318	0	0	0	1
resid	992117	0.068	0.252	0	0	0	1
aadr	992117	69099.476	48051.206	0	49770	78210	11970396
monthly_pension	992117	1743.642	250.771	303	1685	1799	3678
capping	992117	0.759	0.428	0	1	1	1
contrib_m_ind	992117	489.285	67.194	12	492	528	540
contrib_y_ageclass	992117	42.341	3.788	1	43	44	45
splitting	992117	0.798	0.401	0	1	1	1
bonus_m_edu	992117	102.559	52.113	0	96	126	468
bonus_m_assist	992117	0.122	2.686	0	0	0	198
benef_type	992117	1	0	1	1	1	1
marital_stat	992117	3	0	3	3	3	3
scale	992117	42.442	4.205	13	44	44	44
marital_stat1	992117	0	0	0	0	0	0
marital_stat2	992117	0	0	0	0	0	0
marital_stat3	992117	1	0	1	1	1	1
marital_stat4	992117	0	0	0	0	0	0
benef_type1	992117	1	0	1	1	1	1
benef_type2	992117	0	0	0	0	0	0
benef_type3	992117	0	0	0	0	0	0
benef_type4	992117	0	0	0	0	0	0
benef_type5	992117	0	0	0	0	0	0
benef_type6	992117	0	0	0	0	0	0
benef_type7	992117	0	0	0	0	0	0
benef_type8	992117	0	0	0	0	0	0
cluster_id	992117	2	0	2	2	2	2

Table A.3: Summary Statistics of the Clustered Pension Register, Cluster 2

Variable	N	Mean	Std. Dev.	Min	Pctl. 25	Pctl. 75	Max
year	138843	2020	0	2020	2020	2020	2020
age	138843	68.836	18.228	0	65	80	99
sex	138843	0.958	0.202	0	1	1	1
nat	138843	0.959	0.197	0	1	1	1
resid	138843	0.969	0.174	0	1	1	1
aadr	138843	50898.317	95724.49	0	19908	66834	11636226
monthly_pension	138843	324.314	365.315	7	49	474	1969
capping	138843	0.001	0.026	0	0	0	1
contrib_m_ind	138843	97.544	91.727	0	27	140	540
contrib_y_ageclass	138843	37.736	9.003	0	33	44	45
splitting	138843	0.051	0.221	0	0	0	1
bonus_m_edu	138843	26.87	51.354	0	0	30	420
bonus_m_assist	138843	0.008	0.706	0	0	0	132
benef_type	138843	2.814	1.631	2	2	2	8
marital_stat	138843	3.624	0.733	1	4	4	4
scale	138843	9.936	9.203	0	2	15	37
marital_stat1	138843	0.023	0.149	0	0	0	1
marital_stat2	138843	0.083	0.276	0	0	0	1
marital_stat3	138843	0.141	0.348	0	0	0	1
marital_stat4	138843	0.753	0.431	0	1	1	1
benef_type1	138843	0	0	0	0	0	0
benef_type2	138843	0.775	0.417	0	1	1	1
benef_type3	138843	0.037	0.189	0	0	0	1
benef_type4	138843	0.007	0.084	0	0	0	1
benef_type5	138843	0	0.007	0	0	0	1
benef_type6	138843	0.142	0.349	0	0	0	1
benef_type7	138843	0.037	0.189	0	0	0	1
benef_type8	138843	0.002	0.042	0	0	0	1
cluster_id	138843	3	0	3	3	3	3

Table A.4: Summary Statistics of the Clustered Pension Register, Cluster 3

Variable	N	Mean	Std. Dev.	Min	Pctl. 25	Pctl. 75	Max
year	692197	2020	0	2020	2020	2020	2020
age	692197	75.352	7.088	62	70	80	99
sex	692197	0.464	0.499	0	0	1	1
nat	692197	0.927	0.26	0	1	1	1
resid	692197	0.954	0.21	0	1	1	1
aadr	692197	43720.467	56908.96	0	21330	54036	17647020
monthly_pension	692197	415.896	430.957	23	87	608	2558
capping	692197	0.089	0.285	0	0	0	1
contrib_m_ind	692197	112.388	104.372	0	30	164	528
contrib_y_ageclass	692197	42.403	3.53	0	42	44	45
splitting	692197	0.41	0.492	0	0	1	1
bonus_m_edu	692197	25.209	46.073	0	0	36	420
bonus_m_assist	692197	0.008	0.776	0	0	0	144
benef_type	692197	1	0	1	1	1	1
marital_stat	692197	2.94	0.778	1	3	3	4
scale	692197	9.827	9.145	0	3	14	38
marital_stat1	692197	0.091	0.287	0	0	0	1
marital_stat2	692197	0.062	0.241	0	0	0	1
marital_stat3	692197	0.664	0.472	0	0	1	1
marital_stat4	692197	0.184	0.387	0	0	0	1
benef_type1	692197	1	0	1	1	1	1
benef_type2	692197	0	0	0	0	0	0
benef_type3	692197	0	0	0	0	0	0
benef_type4	692197	0	0	0	0	0	0
benef_type5	692197	0	0	0	0	0	0
benef_type6	692197	0	0	0	0	0	0
benef_type7	692197	0	0	0	0	0	0
benef_type8	692197	0	0	0	0	0	0
cluster_id	692197	4	0	4	4	4	4

Table A.5: Summary Statistics of the Clustered Pension Register, Cluster 4

Variable	N	Mean	Std. Dev.	Min	Pctl. 25	Pctl. 75	Max
year	111005	2020	0	2020	2020	2020	2020
age	111005	42.484	24.048	0	18	62	99
sex	111005	0.773	0.419	0	1	1	1
nat	111005	0.246	0.431	0	0	0	1
resid	111005	0.229	0.42	0	0	0	1
aadr	111005	81124.379	86512.036	0	55458	92430	8190720
monthly_pension	111005	1236.74	511.661	129	791	1744	2816
capping	111005	0.019	0.136	0	0	0	1
contrib_m_ind	111005	354.089	133.756	12	252	480	540
contrib_y_ageclass	111005	32.191	11.069	1	24	44	45
splitting	111005	0.33	0.47	0	0	1	1
bonus_m_edu	111005	78.731	66.195	0	12	120	468
bonus_m_assist	111005	0.023	1.368	0	0	0	156
benef_type	111005	3.368	1.999	2	2	4	8
marital_stat	111005	2.907	1.097	1	2	4	4
scale	111005	40.646	6.23	11	40	44	44
marital_stat1	111005	0.075	0.263	0	0	0	1
marital_stat2	111005	0.428	0.495	0	0	1	1
marital_stat3	111005	0.012	0.108	0	0	0	1
marital_stat4	111005	0.485	0.5	0	0	1	1
benef_type1	111005	0	0	0	0	0	0
benef_type2	111005	0.56	0.496	0	0	1	1
benef_type3	111005	0.168	0.374	0	0	0	1
benef_type4	111005	0.053	0.225	0	0	0	1
benef_type5	111005	0	0.015	0	0	0	1
benef_type6	111005	0.012	0.108	0	0	0	1
benef_type7	111005	0.195	0.396	0	0	0	1
benef_type8	111005	0.012	0.109	0	0	0	1
cluster_id	111005	5	0	5	5	5	5

Table A.6: Summary Statistics of the Clustered Pension Register, Cluster 5

Scripts



This appendix presents the most important wrappers and modules used in the package `rrclust` (Lettry 2021).

```
1 #' @title Wrapper to execute the Kamila algorithm.
2 #'
3 #' @description Simple function which executes the computations
4   needed for the
5   Kamila algorithm.
6 #'
7 #' @param tl_inp tidylist of inputs
8 #'
9 #' @return a 'tidylist' containing the following tidylists:
10 #' - 'tl_computation_kamila'
11 #'
12 #' @export
13 #' Last change: 2021-06-17 / Llc
14
15 wrap_kamila_ <- function(tl_inp_kamila) {
16
17   # Dataset preparation
18   tl_prepadata <- wrap_prepadata(tl_inp = tl_inp_kamila)
19
20   # Main computation
21   tl_computation_kamila <- wrap_computation_kamila(
22     tl_inp = tl_inp_kamila,
23     tl_prepadata = tl_prepadata
24   )
25
26   # Output
27   tl_computation_kamila
28 }
29
30
31 #' @title wrap_kamila (memoised)
32 #' @export
33 wrap_kamila <- memoise::memoise(wrap_kamila_)
```

Script B.1: Wrapper for the data preparation and the Kamila algorithm execution

```

1 #' @title Wrapper for the preparation of the data
2 #'
3 #' @description This wrapper contains all the necessary modules
4   which allow to
5   #' prepare the data.
6   #'
7   #' @param tl_inp List of input data frames.
8   #'
9   #' @return a 'tidylist' containing the following tidylists:
10  #' - 'tl_prepa_rr'
11  #'
12  #' @author [Layal Christine Lettry](mailto:layalchristine.
13    lettry@unifr.ch)
14  #'
15  #' @export
16
17  # Last change: 2021-02-25 / Llc
18
19  wrap_prepadata_ <- function(tl_inp) {
20
21    # Register of rents
22
23    tl_prepa_rr <- mod_prepa_rr(
24      IND_YEARLY_RR = tl_inp$IND_YEARLY_RR
25    )
26
27    # Training and validation sets
28
29    tl_mod_tsvs <- mod_tsvs(
30      RR_OASI = tl_prepa_rr$RR_OASI,
31      PARAM_GLOBAL = tl_inp$PARAM_GLOBAL
32    )
33
34    # Full datasets, training and validation sets of categorical
35    # and continuous
36    # variables
37
38    tl_mod_catcontvar <- mod_catcontvar(
39      RR_OASI = tl_prepa_rr$RR_OASI,
40      RR_OASI_TS = tl_mod_tsvs$RR_OASI_TS,
41      RR_OASI_VS = tl_mod_tsvs$RR_OASI_VS,
42      PARAM_GLOBAL = tl_inp$PARAM_GLOBAL
43    )
44  }

```

```

43 |   # Output
44 |   c(tl_mod_catcontvar)
45 | }

48 | #' @title wrap_prepadata (memoised)
49 | #' @export
50 | wrap_prepadata <- memoise::memoise(wrap_prepadata_)

```

Script B.2: Wrapper for the preparation of the data

```

1 #' @title Preparation of the register of rents data.
2 #'
3 #' @description Prepares the variables of the register of rents
4 #'
5 #' @param IND_YEARLY_RR a data frame containing the data of the
6 #'   register of rents
7 #'   subsetted for one year only.
8 #'
9 #' @param list List of input data frames.
10 #'
11 #' @return a 'tidylist' containing the following tidy data
12 #'   frames:
13 #'   - 'RR_OASI' : contains all the beneficiaries of the OASI.
14 #'
15 #' @author [Layal Christine Lettry](mailto:layalchristine.
16 #'   lettry@unifr.ch)
17 #'
18 #' @export
19
20 # - 'Last change': 2021-09-02 / Llc
21
22 mod_prepa_rr <- function(IND_YEARLY_RR,
23                           list = NULL) {
24   mod_init()
25
26   # --- Recoding and renaming the variables
27   -----
28   RR_OASI1 <- if ("lbedu" %in% names(IND_YEARLY_RR)) {
29     IND_YEARLY_RR %>%
30       dplyr::rename(
31         "aadr" = ram,
32         "monthly_pension" = monatliche_rente,
33         "age" = alt,
34         "year" = jahr,
35         "resid" = dom,
36         "contrib_m_ind" = lcot,
37         "contrib_y_ageclass" = lcotg,
38         "splitting" = csplit,
39         "bonus_m_edu" = lbedu,
40         "bonus_m_assist" = lbass,
41         "capping" = cplaf
42       ) %>%
43       mutate(
44         # Recode contrib_m_ind for NA values

```

```

41     contrib_m_ind = case_when(
42       is.na(contrib_m_ind) ~ as.double(-0),
43       TRUE ~ as.double(contrib_m_ind)
44     ),
45     # Recode contrib_y_ageclass for NA values
46     contrib_y_ageclass = case_when(
47       is.na(contrib_y_ageclass) ~ as.double(-0),
48       TRUE ~ as.double(contrib_y_ageclass)
49     ),
50     # Recode splitting for NA values
51     splitting = case_when(
52       is.na(splitting) ~ as.double(-0),
53       TRUE ~ as.double(splitting)
54     ),
55     # Recode bonus_m_edu for NA values
56     bonus_m_edu = case_when(
57       is.na(bonus_m_edu) ~ as.double(-0),
58       TRUE ~ as.double(bonus_m_edu)
59     ),

61     # Recode bonus_m_assist for NA values
62     bonus_m_assist = case_when(
63       is.na(bonus_m_assist) ~ as.double(-0),
64       TRUE ~ as.double(bonus_m_assist)
65     )
66   )
67 } else {
68   IND_YEARLY_RR %>%
69     dplyr::rename(
70       "aadr" = ram,
71       "monthly_pension" = monatliche_rente,
72       "age" = alt,
73       "year" = jahr,
74       "resid" = dom
75     )
76 }

79 # Rename the realizations of the variables
80 RR_OASI2 <- RR_OASI1 %>%
81   mutate(
82     benef_type = dplyr::recode(gpr,
83       "rvieillesse_simple" = 1, # "Old-age"
84       "rveuve" = 2, # "Widow"
85       "rorphelin_pere_simple" = 3, # "Father's orphan"

```



```

86     "rorphelin_mere_simple" = 4, # "Mother's orphan"
87     "rorphelin_double" = 5, # "Twice orphan",
88     "rcompl_femme" = 6, # "Spouse's compl.",
89     "renfant_pere_simple" = 7, # "Father's child rent"
90     "renfant_mere_simple" = 8 # "Mother's child rent"
91   ),
92   sex = dplyr::recode(sex,
93     "f" = 1, # "Woman"
94     "m" = 0 # "Man"
95   ),
96   nat = dplyr::recode(nat,
97     "au" = 1, # "Foreign",
98     "ch" = 0 # "Swiss"
99   ),
100  resid = dplyr::recode(resid,
101    "au" = 1, # "Foreign",
102    "ch" = 0, # "Swiss"
103  ),
104  marital_stat = dplyr::recode(zv,
105    "geschieden" = 1, # "Divorced",
106    "ledig" = 2, # "Single",
107    "verheiratet" = 3, # "Married",
108    "verwitwet" = 4 # "Widowed"
109  ),
110  # Mutate the total years of contribution
111  scale = round(eprc * 44, 0),
112  ) %>%
113  # Transform character variables to factors
114  mutate_if(sapply(., is.character), as.factor) %>%
115  mutate(
116    # Dummy variables for each marital status
117    marital_stat1 = case_when(marital_stat == 1 ~ 1, TRUE ~
118      0),
119    marital_stat2 = case_when(marital_stat == 2 ~ 1, TRUE ~
120      0),
121    marital_stat3 = case_when(marital_stat == 3 ~ 1, TRUE ~
122      0),
123    marital_stat4 = case_when(marital_stat == 4 ~ 1, TRUE ~
124      0),
125
126    # Dummy variables for each benefit type
127    benef_type1 = case_when(benef_type == 1 ~ 1, TRUE ~ 0),
128    benef_type2 = case_when(benef_type == 2 ~ 1, TRUE ~ 0),
129    benef_type3 = case_when(benef_type == 3 ~ 1, TRUE ~ 0),
130    benef_type4 = case_when(benef_type == 4 ~ 1, TRUE ~ 0),

```

```
127     benef_type5 = case_when(benef_type == 5 ~ 1, TRUE ~ 0),
128     benef_type6 = case_when(benef_type == 6 ~ 1, TRUE ~ 0),
129     benef_type7 = case_when(benef_type == 7 ~ 1, TRUE ~ 0),
130     benef_type8 = case_when(benef_type == 8 ~ 1, TRUE ~ 0),

132     # Recode age_ret for NA values
133     age_retire = case_when(
134       is.na(age_ret) ~ as.double(-99999),
135       TRUE ~ as.double(age_ret)
136     ),
137   ) %>%
138   dplyr::select(
139     -zv,
140     -gpr
141   )

143   # Verify all variables are numeric
144   RR_OASI <- if ("napref" %in% names(RR_OASI2)) {
145     RR_OASI2 %>%
146       dplyr::select(
147         -napref
148       ) %>%
149       mutate_all(funs(as.numeric(.)))
150   } else {
151     RR_OASI2 %>%
152       mutate_all(funs(as.numeric(.)))
153   }

155   mod_return(
156     RR_OASI
157   )
158 }
```

Script B.3: Preparation of the RR_OASI tibble

```

1  #' @title Splitting the data into a Training and a Validation
   sets
2  #'
3  #' @description Splits the data into a training and a
   validation sets, given
4  #' the too large number of observations, in order to determine
   the best number of
5  #' clusters for the KAMILA algorithm.
6  #'
7  #' @param RR_OASI a data frame containing the all the data.
8  #'
9  #' @param PARAM_GLOBAL a data frame containing the parameters.
   We use the following:
10 #' - 'pct_sample_ts': percentage of observations which build
   the training set.
11 #'
12 #' @param list List of input data frames.
13 #'
14 #' @return a 'tidylist' containing the following tidy data
   frames:
15 #' - 'RR_OASI_TS' : Training set of categorical data.
16 #' - 'RR_OASI_VS': Validation set of categorical data.
17 #'
18 #' @references [www.geeksforgeeks.org](https://www.
   geeksforgeeks.org/the-validation-set-approach-in-r-
   programming/)
19 #' @author [Layal Christine Lettry](mailto:layalchristine.
   lettry@unifr.ch)
20 #' @import caTools
21 #' @export
22
23 # - 'Last change': 2022-08-11 / Llc
24
25 mod_tsvs <- function(RR_OASI,
26                       PARAM_GLOBAL,
27                       list = NULL) {
28   mod_init()
29
30   # Setting seed to generate a reproducible random sampling
31   set.seed(100)
32
33   # Choose a categorical balanced variable to do the splitting:
   sex and check
34   # if it is balanced
35   freqtable <- table(RR_OASI$sex)

```

```
36 | proptable <- prop.table(freqtable) # approximately balanced
38 | # Dividing the complete RR_OASI dataset into 2 parts having
    |   ratio of 99.9% and 0.1%
39 | oasi_spl <- sample.split(RR_OASI$sex,
40 |   SplitRatio = PARAM_GLOBAL$pct_sample_ts / 100
41 | )
43 | # Training set
44 | # Selecting that part of RR_OASI dataset which belongs to the
    |   0.1% of the dataset
45 | # divided in previous step
46 | RR_OASI_TS <- subset(RR_OASI, oasi_spl == TRUE)
48 | # Validation set
49 | # Selecting that part of RR_OASI dataset which belongs to the
    |   99.9% of the dataset
50 | # divided in previous step
51 | RR_OASI_VS <- subset(RR_OASI, oasi_spl == FALSE)
53 | # checking number of rows and column in training and
    |   validation datasets
54 | print(dim(RR_OASI_TS))
55 | print(dim(RR_OASI_VS))
57 | mod_return(
58 |   RR_OASI_TS,
59 |   RR_OASI_VS
60 | )
61 | }
```

Script B.4: Training set and validation set

```

1  #' @title Splitting the continuous from the categorical data
2  #'
3  #' @description Splits the continuous from the categorical data
4  #' , in order to use
5  #' a clustering method.
6  #'
7  #' @param RR_OASI a data frame containing the all the data,
8  #' whose variables are:
9  #' - 'year': Year of the pension register extract.
10 #' - 'age': Age of the individual.
11 #' - 'age_retire': Retirement age.
12 #' - 'sex': Sex, if 1: female, if 0: male
13 #' - 'nat': Nationality, if 1: Foreign, if 0: Swiss.
14 #' - 'resid': Residence, if 1: Foreign, if 0: Swiss.
15 #' - 'benef_type1': If 1, Old-age type of benefit (dummy)
16 #' - 'benef_type2': Widow type of benefit (dummy)
17 #' - 'benef_type3': Father's orphan type of benefit (dummy)
18 #' - 'benef_type4': Mother's orphan type of benefit (dummy)
19 #' - 'benef_type5': Twice orphan type of benefit (dummy)
20 #' - 'benef_type6': Spouse's compl. type of benefit (dummy)
21 #' - 'benef_type7': Father's child rent type of benefit (dummy
22 #' )
23 #' - 'benef_type8': Mother's child rent type of benefit (dummy
24 #' )
25 #' - 'benef_type': Types of benefits type of benefit (
26 #' categorical)
27 #' - 'marital_stat1': Divorced marital status (dummy)
28 #' - 'marital_stat2': Single as reference category marital
29 #' status (dummy)
30 #' - 'marital_stat3': Married marital status (dummy)
31 #' - 'marital_stat4': Widowed marital status (dummy)
32 #' - 'marital_stat': Marital Status
33 #' - 'splitting': If 1, splitting of the revenues, 0 otherwise
34 #' .
35 #' - 'capping': If 1, the pension is capped, 0 otherwise.
36 #' - 'contrib_m_ind': total number of OASI contribution months
37 #' per individual.
38 #' - 'contrib_y_ageclass': total number of contribution years
39 #' per age group.
40 #' - 'bonus_m_edu': number of months paid with a bonus for
41 #' educative tasks.
42 #' - 'bonus_m_assist': number of months paid with a bonus for
43 #' assistance/care
44 #' tasks.
45 #'

```

```

35 #' @param RR_OASI_TS a training dataset containing x% of the
    data.
36 #'
37 #' @param RR_OASI_VS a validation dataset containing (100 - x)%
    of the data.
38 #'
39 #' @param PARAM_GLOBAL a data frame containing the parameters.
    We use the following:
40 #' - 'categ_var': Chosen categorical variables
41 #' - 'cont_var': Chosen continuous variables
42 #'
43 #' @param list List of input data frames.
44 #'
45 #' @return a 'tidylist' containing the following tidy data
    frames:
46 #' - 'CATEG_DF': contains only categorical variables (factors
    )
47 #' - 'CONT_DF' : contains only continuous variables (numeric)
48 #' - 'CATEG_DF_TS': contains only categorical variables (
    factors), training set
49 #' - 'CONT_DF_TS' : contains only continuous variables (
    numeric), training set
50 #' - 'CATEG_DF_VS': contains only categorical variables (
    factors), validation set
51 #' - 'CONT_DF_VS' : contains only continuous variables (
    numeric), validation set
52 #'
53 #' @references [www.geeksforgeeks.org](https://www.
    geeksforgeeks.org/the-validation-set-approach-in-r-
    programming/)
54 #' @author [Layal Christine Lettry](mailto:layalchristine.
    lettry@unifr.ch)
55 #' @export
56
57 # - 'Last change': 2022-08-11 / Llc
58
59 mod_catcontvar <- function(RR_OASI,
60                             RR_OASI_TS,
61                             RR_OASI_VS,
62                             PARAM_GLOBAL,
63                             list = NULL) {
64   mod_init()
65
66   # Chosen categorical variables
67   categ_var <- separate_at_comma(PARAM_GLOBAL$categ_var)

```

```

69  # Chosen continuous variables
70  cont_var ← separate_at_comma(PARAM_GLOBAL$cont_var)

72  #--- Full dataset
    -----

73  # Dataframe of categorical variables
74  CATEG_DF ← RR_OASI %>%
75    dplyr::select(any_of(categ_var)) %>%
76    # Transform all variables as factors
77    mutate_all(as.factor)

79  # Dataframe of continuous variables
80  CONT_DF ← RR_OASI %>%
81    dplyr::select(any_of(cont_var)) %>%
82    # Transform all variables as numeric
83    mutate_all(as.numeric)

85  # checking number of rows and column training datasets
86  print(dim(CATEG_DF))
87  print(dim(CONT_DF))

89  #--- Training set
    -----

90  # Dataframe of categorical variables
91  CATEG_DF_TS ← RR_OASI_TS %>%
92    dplyr::select(any_of(categ_var)) %>%
93    # Transform all variables as factors
94    mutate_all(as.factor)

96  # Dataframe of continuous variables
97  CONT_DF_TS ← RR_OASI_TS %>%
98    dplyr::select(any_of(cont_var)) %>%
99    # Transform all variables as numeric
100    mutate_all(as.numeric)

102  # checking number of rows and column training datasets
103  print(dim(CATEG_DF_TS))
104  print(dim(CONT_DF_TS))

106  #--- Validation set
    -----

```

```
107 | # Dataframe of categorical variables
108 | CATEG_DF_VS <- RR_OASI_VS %>%
109 |   dplyr::select(any_of(categ_var)) %>%
110 |   # Transform all variables as factors
111 |   mutate_all(as.factor)

113 | # Dataframe of continuous variables
114 | CONT_DF_VS <- RR_OASI_VS %>%
115 |   dplyr::select(any_of(cont_var)) %>%
116 |   # Transform all variables as numeric
117 |   mutate_all(as.numeric)

119 | # checking number of rows and column validation datasets
120 | print(dim(CATEG_DF_VS))
121 | print(dim(CONT_DF_VS))

123 | mod_return(
124 |   CATEG_DF,
125 |   CONT_DF,
126 |   CATEG_DF_TS,
127 |   CONT_DF_TS,
128 |   CATEG_DF_VS,
129 |   CONT_DF_VS
130 | )
131 | }
```

Script B.5: Splitting the variables according to their types


```

1 #' @title Wrapper for the clusters construction.
2 #'
3 #' @description This wrapper contains all the necessary modules
4 #'   which allow to
5 #'   construct the clusters.
6 #'
7 #' @param tl_inp List of input data frames of which we use:
8 #' - 'PARAM_KAMILA$calc_kstar': If TRUE, estimates the clusters
9 #'   . Else, takes the
10 #'   parameter PARAM_KAMILA$param_kstar.
11 #' - 'PARAM_KAMILA$cont_var_expl': List of continuous variables
12 #'   chosen as explicative
13 #'   variables.
14 #' - 'PARAM_KAMILA$categ_var_expl': List of categorical
15 #'   variables chosen as explicative
16 #'   variables.
17 #'
18 #' @param tl_prepadata List of data frames prepared in a first
19 #'   step.
20 #'
21 #' @return a 'tidylist' containing the following tidylists:
22 #' - 'tl_mod_calc_kamila'
23 #'
24 #' @author [Layal Christine Lettry](mailto:layalchristine.
25 #'   lettry@unifr.ch)
26 #'
27 #' @export
28
29 # Last change: 2021-09-02 / Llc
30
31 wrap_computation_kamila_ <- function(tl_inp,
32                                     tl_prepadata) {
33
34   # Select the desired continuous explicative variables
35   cont_var_expl <- separate_at_comma(tl_inp$PARAM_KAMILA$cont_
36                                     var_expl)
37
38   CONT_DF_TS <- tl_prepadata$CONT_DF_TS %>%
39     dplyr::select(any_of(cont_var_expl))
40
41   CONT_DF_VS <- tl_prepadata$CONT_DF_VS %>%
42     dplyr::select(any_of(cont_var_expl))
43
44   CONT_DF <- tl_prepadata$CONT_DF %>%

```

```

39     dplyr::select(any_of(cont_var_expl))

42     # Select the desired categorical explicative variables
43     categ_var_expl ← separate_at_comma(tl_inp$PARAM_KAMILA$categ_
      var_expl)

45     CATEG_DF_TS ← tl_prepadata$CATEG_DF_TS %>%
46     dplyr::select(any_of(categ_var_expl))

48     CATEG_DF_VS ← tl_prepadata$CATEG_DF_VS %>%
49     dplyr::select(any_of(categ_var_expl))

51     CATEG_DF ← tl_prepadata$CATEG_DF %>%
52     dplyr::select(any_of(categ_var_expl))

55     # Run the algorithm on the TS to find the optimal number of
      clusters kstar
56     # Writes kstar as the parameter PARAM_KAMILA$param_kstar
57     if (tl_inp$PARAM_KAMILA$calc_kstar) {
58       tl_mod_kstar ← mod_kstar(
59         PARAM_KAMILA = tl_inp$PARAM_KAMILA,
60         CATEG_DF_TS = CATEG_DF_TS,
61         CONT_DF_TS = CONT_DF_TS
62       )
63       KM_RES ← tl_mod_kstar$KM_RES
64     } else {
65       KM_RES ← tibble(cluster_id = NA_real_) %>%
66       mutate(
67         kstar = NA_real_,
68         ps_values = NA_real_, # Prediction Strength value
69         avg_pred_str = NA_real_, # Average prediction strength
70         std_err_pred_str = NA_real_, # SE pred. strength
71         ps_cv_res_run = NA_real_, # Pred. Strength CV residuals
72         cluster_id = NA_real_
73       )
74     }

76     # PARAM_KAMILA with the updated kstar parameter
77     PARAM_KAMILA ← if (tl_inp$PARAM_KAMILA$calc_kstar) {
78       tl_mod_kstar$PARAM_KAMILA
79     } else {
80       tl_inp$PARAM_KAMILA
81     }

```

```

84   # Apply kstar to the whole dataset
85   tl_mod_calc_kamila ← mod_calc_kamila(
86     PARAM_KAMILA = PARAM_KAMILA,
87     CONT_DF = CONT_DF,
88     FULL_CONT_DF = tl_prepadata$CONT_DF,
89     CATEG_DF = CATEG_DF,
90     FULL_CATEG_DF = tl_prepadata$CATEG_DF,
91     KM_RES = KM_RES
92   )

94   # Output
95   if (tl_inp$PARAM_KAMILA$calc_kstar) {
96     c(
97       tl_mod_calc_kamila,
98       tl_mod_kstar
99     )
100  } else {
101    c(tl_mod_calc_kamila)
102  }
103 }

106 #' @title wrap_computation_kamila (memoised)
107 #' @export
108 wrap_computation_kamila ← memoise::memoise(wrap_computation_
      kamila_)

```

Script B.6: Wrapper for the Kamila algorithm execution part

```

1 #' @title Estimation of the best number of clusters using the
   #' Kamila algorithm.
2 #'
3 #' @description Estimation of best number of clusters using the
   #' Kamila algorithm
4 #' on the training set.
5 #'
6 #' @param PARAM_KAMILA dataframe with all needed parameters for
   #' the Kamila method,
7 #' from which the following parameters are used:
8 #' - 'numberofclusters': The number of clusters returned by the
   #' algorithm, i.e.
9 #' sequence indicating the number of clusters which should be
   #' investigated to
10 #' extract the optimal number of clusters.
11 #' - 'numinit': The number of initializations used.
12 #' - 'maxiter': The maximum number of iterations in each run.
13 #' - 'calcnumclust': Character: Method for selecting the number
   #' of clusters. Setting
14 #' calcNumClust to ps uses the prediction strength method of
15 #' Tibshirani & Walther (J. of Comp. and Graphical Stats. 14(3),
   #' 2005).
16 #' - 'pred_threshold': Threshold fixed to 0.8 for well
   #' separated clusters (i.e.
17 #' not overlapping).
18 #'
19 #' @param CATEG_DF_TS Training set of the register of rents
   #' containing all categorical
20 #' variables as factors.
21 #'
22 #' @param CONT_DF_TS Training set of the register of rents
   #' containing all the continuous
23 #' variables.
24 #'
25 #' @return a tidylist containing the following tidy data frames
   #' :
26 #' - 'KM_RES' database containing the results of the
   #' clustering.
27 #' - 'PARAM_KAMILA' dataframe with the updated kstar parameter
   #' .
28 #'
29 #' @author [Layal Christine Lettry](mailto:layalchristine.
   #' lettry@unifr.ch)
30 #'
31 #' @export

```

```

32  #' @import kamila
34  # Last change: 2021-06-17 / Llc
36  mod_kstar <- function(PARAM_KAMILA,
37                        CATEG_DF_TS,
38                        CONT_DF_TS,
39                        list = NULL) {
40    mod_init()

43    #--- 1.1) Standardize the continuous variables
        -----

45    CONTVARS <- as.data.frame(lapply(CONT_DF_TS, rangeStandardize)
46                               )
47    names(CONTVARS) <- paste0(names(CONTVARS), "_std")
48    CATFACTOR <- as.data.frame(CATEG_DF_TS)

50    #--- 1.2) Estimate the best number of clusters g*
        -----

52    # Computes the clusters and reruns the inputs with the newest
        results.

54    # Setting seed to generate a reproducible random sampling
55    set.seed(6)

57    # Number of clusters to be returned by the algorithm
58    numberofclusters <- as.numeric(eval(parse(
59      text =
60        PARAM_KAMILA$numberofclusters
61      )))

63    # Running the algorithm on the Training Set
64    kmresps <- kamila(
65      conVar = CONTVARS,
66      catFactor = CATFACTOR,
67      numClust = numberofclusters,
68      numInit = PARAM_KAMILA$numinit,
69      maxIter = PARAM_KAMILA$maxiter,
70      calcNumClust = PARAM_KAMILA$calcnumclust,
71      predStrThresh = PARAM_KAMILA$pred_threshold
72    )

```

```

75  # Optimal number of clusters
76  KSTAR ← tibble(cluster_id = as.integer(names(kmresps$nClust$
77    psValues))) %>%
    mutate(kstar = kmresps$nClust$bestNClust)

79  # Other information of the run
80  # Note: PS = 1 - Variance
81  NCLUST ← tibble(cluster_id = as.integer(names(kmresps$nClust$
82    psValues))) %>%
    mutate(
83      ps_values = kmresps$nClust$psValues, # Prediction
        strength value
84      avg_pred_str = kmresps$nClust$avgPredStr, # Average
        prediction strength
85      std_err_pred_str = kmresps$nClust$stdErrPredStr # SE pred
        . strength
86    )

89  PS_CV_RES ← kmresps$nClust$psCvRes %>% # Pred. Strength CV
    residuals
90  as_tibble() %>%
91  mutate(cluster_id = as.integer(rownames(kmresps$nClust$
92    psCvRes)))
93  colnames(PS_CV_RES)[!grepl(
94    "cluster_id",
95    colnames(PS_CV_RES)
96  )] ← paste("ps_cv_res_run",
97    1:PARAM_KAMILA$numinit,
98    sep = "_"
99  )

100 # Join all datasets of results
101 KM_RES ← KSTAR %>%
102   left_join(NCLUST,
103     by = "cluster_id"
104   ) %>%
105   left_join(PS_CV_RES,
106     by = "cluster_id"
107   )

109 # Save the optimal number of clusters in a parameter
110 PARAM_KAMILA$param_kstar ← kmresps$nClust$bestNClust

```

```
113 | mod_return(  
114 |     KM_RES ,  
115 |     PARAM_KAMILA  
116 | )  
117 | }
```

Script B.7: Finding the parameter kstar

```

1 #' @title Splitting the initial dataset into kstar clusters.
2 #'
3 #' @description Splitting the initial dataset into kstar
4 #' clusters by using the
5 #' parameter kstar determined in the module \code{\link{mod_
6 #' kstar}}.
7 #'
8 #' @param PARAM_KAMILA dataframe with all needed parameters for
9 #' the Kamila method,
10 #' from which the following parameters are used:
11 #' - 'nunit': The number of initializations used.
12 #' - 'maxiter': The maximum number of iterations in each run.
13 #' - 'param_kstar': Best number of clusters estimated in the
14 #' module
15 #' \code{\link{mod_kstar}}.
16 #'
17 #' @param CATEG_DF subset of the register of rents containing
18 #' all categorical
19 #' variables as factors except for the nominal variables
20 #' marital_stat and benef_type.
21 #'
22 #' @param CONT_DF subset of the register of rents containing
23 #' all the continuous
24 #' variables except for the outcome variables aadr and monthly_
25 #' pension.
26 #'
27 #' @param FULL_CONT_DF database containing the continuous
28 #' variables used for the
29 #' estimation plus the outcome variables aadr and monthly_
30 #' pension.
31 #'
32 #' @param FULL_CATEG_DF database containing the categorical
33 #' variables used for
34 #' the estimation plus the nominal variables marital_stat and
35 #' benef_type.
36 #'
37 #' @return a 'tidylist' containing the following tidy data
38 #' frames:
39 #' - 'PLOTDATKAM' database containing the clusters factor and
40 #' the other
41 #' variables.
42 #' - 'KM_RES_FINAL' database containing the resulting
43 #' parameters of the
44 #' clustering.
45 #' - 'CONTVARS' database containing the continuous standardised

```



```

    variables.
31 #' - 'FULL_CONT_DF' database containing the continuous
    variables used for the
32 #' estimation.
33 #' - 'FULL_CATEG_DF' database containing the categorical
    variables used for
34 #' the estimation.
35 #'
36 #' @author [Layal Christine Lettry](mailto:layalchristine.
    lettry@unifr.ch)
37 #' @export
38 #' @import kamila

40 # Last change: 2021-09-02 / Llc

42 mod_calc_kamila <- function(PARAM_KAMILA,
43                             CONT_DF,
44                             CATEG_DF,
45                             FULL_CONT_DF,
46                             FULL_CATEG_DF,
47                             KM_RES,
48                             list = NULL) {
49   mod_init()

52   #--- 1.1 Construction of the g* (from 1.2) clusters with the
    Kamila method---

54   CONTVARS <- as.data.frame(lapply(CONT_DF, rangeStandardize))
55   names(CONTVARS) <- paste0(names(CONTVARS), "_std")

57   CATFACTOR <- as.data.frame(CATEG_DF)

59   #--- 1.2 Construction of the g* (from 1.2) clusters with the
    Kamila method on
60   # the whole dataset
    -----

62   # Setting seed to generate a reproducible random sampling
63   set.seed(5)

65   kstar <- PARAM_KAMILA$param_kstar

67   kmres <- kamila(
68     conVar = CONTVARS,

```

```

69     catFactor = CATFACTOR,
70     numClust = kstar,
71     numInit = PARAM_KAMILA$numinit,
72     maxIter = PARAM_KAMILA$maxiter
73 )

75 # Transform the number of clusters into factors
76 cluster_id ← factor(kmres$finalMemb)

78 # Retrieve all clustering estimation resulting parameters
79 KM_RES_FINAL ← tibble(
80     final_loglik = kmres$finalLogLik,
81     final_obj = kmres$finalObj,
82     num_clust = kmres$input$numClust,
83     max_iterations = kmres$input$maxIter,
84     categorical_bw = kmres$input$catBw
85 )

87 # Construction of a Dataframe for plotting the estimation
results
88 PLOTDAKAM ← cbind(
89     cluster_id,
90     CONTVARS,
91     FULL_CONT_DF,
92     # CATFACTOR,
93     FULL_CATEG_DF
94 ) %>%
95     as_tibble()

97 mod_return(
98     PLOTDAKAM,
99     CONTVARS,
100     FULL_CONT_DF,
101     FULL_CATEG_DF,
102     KM_RES_FINAL
103 )
104 }

```

Script B.8: Splitting the register of rents into kstar clusters

```
1 #' @title Function writing the packages version used to run
  this output
2 #' @description Writes the version of the rrclust and dplyr
  packages and the time
3 #' of the output production.
4 #' @param list tidylist
5 #' @author [Layal Christine Lettry](mailto:layalchristine.
  lettry@unifr.ch)
6 #' @return 'LOG' tibble with information about time and
  packages version.
7 #' @export

9 # Last change: 2021-06-17 / Llc

11 mod_log <- function(list = NULL) {
12   LOG <- tibble(
13     rrclust_version = as.character(packageVersion("rrclust")),
14     dplyr_version = as.character(packageVersion("dplyr")),
15     runtime = as.character(Sys.time())
16   )

18   mod_return(LOG)
19 }
```

Script B.9: Log of the executed run

Parameters Files



This appendix presents the most important parameters files used in the package `rrclust` (Lettry 2021).

Table C.1: rrclust parameters file PARAM_KAMILA

Parameter	Values	Description
calc_kstar	TRUE	If TRUE, compute the parameter k^* by means of the function kamila.
numberofclusters	2:20	Give the range of the number of clusters which can be returned by the algorithm (cf. numClust of the function kamila).
numinit	10	Give the number of initialisations.
maxiter	50	For each run, give the number of iterations.
calcnumclust	ps	If calcnumclust==ps, use the prediction strength criterion to select the number of clusters.
pred_threshold	0.8	If calcnumclust == ps, give pred_threshold in order to indicate the threshold for the prediction strength criterion .
param_kstar		If non empty, means that the parameter k^* has already been computed and is given directly by param_kstar in order to skip the step of estimating k^* . In our case, it would be 5 (cf. chapter 3).
categ_var_expl	sex, nat, resid, benef_type1, marital_stat1, marital_stat3, marital_stat4	Give the list of the categorical variables used to determine k^* . They are stored in a tibble (namely CATEG_DF_TS) of these categorical variables enters the parameter catFactor of the kamila function.
cont_var_expl	age, age_retire, scale	Give the list of the continuous variables used to determine k^* . They are stored in a tibble (namely CONT_DF_TS) of these categorical variables enters the parameter catFactor of the kamila function.

Table C.2: rrclust parameters file PARAM_GLOBAL

Parameter	Values	Description
method_name	kamila	Give the method to apply for clustering.
path_data	...	Give the directory path for the input data.
path_data_local	...	Give the directory path for the input data (locally).
path_data_server	...	Give the directory path for the input data (on the server).
description	Kamila method	Give the description of the clustering method.
pct_sample_ts	0.1	Give the percentage of the observations to be stored in the training sample.
gdpath	...	Give the directory to store the figures.
categ_var	sex, nat, resid, benef_type1, benef_type2, benef_type3, benef_type4, benef_type5, benef_type6, benef_type7, benef_type8, benef_type, marital_stat1, marital_stat2, marital_stat3, marital_stat4, marital_stat, splitting, capping	Give the list of the categorical variables in order to be stored in the tibbles CATEG_DF, CATEG_DF_TS and CATEG_DF_VS in the module mod_catcontvar (cf. script B.5).
cont_var	year, aadr, monthly_pension, age, age_retire, scale, contrib_m_ind, contrib_y_ageclass, bonus_m_edu, bonus_m_assist	Give the list of the continuous variables in order to be stored in the tibbles CONT_DF, CONT_DF_TS and CONT_DF_VS in the module mod_catcontvar (cf. script B.5).

Acronyms

AADR	Annual Average Determinant Revenue 5, 6, 9, 10, 19–21, 23, 25, 48–53, 55, 58, 80–85
ARI	Adjusted Rand Index 34
CCO	Central Compensation Office 13, 15, 17, 19, 20, 29, 30, 33, 43, 46, 47, 55
CLARA	clustering large applications 32
DI	Disability Insurance 19–22
FSIO	Federal Social Insurance Office 13, 15, 17, 19, 20, 29, 30, 33, 43, 46, 47, 55
KAMILA	KAMILA clustering method 13, 15, 17, 25, 30, 33, 34, 36, 38–41, 43, 46, 47, 55
KD	Kernel Density 33
LC	latent class 31, 32
LCC	latent class clustering 31
OASI	Old-Age and Survivors' Insurance 6, 9, 15, 17, 19–23, 25, 41, 47, 48, 55, 58
PAM	partitioning around medoids 31, 32
PR	Pension Register 1, 5, 13, 15, 17, 19, 20, 22–26, 29, 30, 32–34, 36, 38–41, 43, 44, 46–48, 50, 52, 55, 58, 60, 62, 64, 66, 68, 70, 72, 74, 76, 78, 80, 82, 84, 86, 88, 90, 92, 94, 96, 98, 100, 102, 104, 106, 108, 110, 112, 114, 116, 118, 120, 122, 124, 126, 128, 132

PS criterion prediction strength criterion 36, 37, 45, 125

Bibliography

Printed References

- Chu, Cheng-tao et al. (2007): “Map-Reduce for Machine Learning on Multicore”. In: B. Schölkopf, J. Platt, and T. Hoffman, eds., *Advances in Neural Information Processing Systems*. Vol. 19. MIT Press. URL: <https://proceedings.neurips.cc/paper/2006/file/77e3bc58ce560b86c2b59363281e914-Paper.pdf>.
- Foss, Alex et al. (July 2016): “A semiparametric method for clustering mixed data”. In: *Machine Learning* 105.3, pp. 419–458. DOI: [10.1007/s10994-016-5575-7](https://doi.org/10.1007/s10994-016-5575-7). URL: <https://doi.org/10.1007/s10994-016-5575-7>.
- Foss, Alexander H. and Marianthi Markatou (2018a): “kamila: Clustering Mixed-Type Data in R and Hadoop”. In: *Journal of Statistical Software, Articles* 83.13, pp. 1–44. ISSN: 1548-7660. DOI: [10.18637/jss.v083.i13](https://www.jstatsoft.org/v083/i13). URL: <https://www.jstatsoft.org/v083/i13>.
- (2018b): “kamila: Clustering Mixed-Type Data in R and Hadoop”. In: *Journal of Statistical Software* 83.13, pp. 1–45. DOI: [10.18637/jss.v083.i13](https://www.jstatsoft.org/v083/i13).
- Gower, J. C. (1971): “A General Coefficient of Similarity and Some of Its Properties”. In: *Biometrics* 27.4, pp. 857–871. ISSN: 0006341X, 15410420. URL: <http://www.jstor.org/stable/2528823>.
- Hennig, Christian and Tim F. Liao (2013): “How to find an appropriate clustering for mixed-type variables with application to socio-economic stratification”. In: *Journal of the Royal Statistical Society: Series C (Applied Statistics)* 62.3, pp. 309–369. ISSN: 1467-9876. DOI: [10.1111/j.1467-9876.2012.01066.x](https://doi.org/10.1111/j.1467-9876.2012.01066.x).
- Huang, Zhexue (1998): “Extensions to the k-Means Algorithm for Clustering Large Data Sets with Categorical Values”. In: *Data Mining and Knowledge Discovery* 2.3, pp. 283–304. ISSN: 1384-5810. DOI: [10.1023/a:1009769707641](https://doi.org/10.1023/a:1009769707641).
- Hunt, Lynette and Murray Jorgensen (July 2011): “Clustering Mixed Data”. In: *Wiley Interdisc. Rev.: Data Mining and Knowledge Discovery* 1, pp. 352–361. DOI: [10.1002/widm.33](https://doi.org/10.1002/widm.33).
- Kaufman, Leonard and Peter Rousseeuw (Jan. 1990): *Finding Groups in Data: An Introduction To Cluster Analysis*. ISBN: 0-471-87876-6. DOI: [10.2307/2532178](https://doi.org/10.2307/2532178).

- Kotz, S., N. Balakrishnan, and N.L. Johnson (2004): *Continuous Multivariate Distributions, Volume 1: Models and Applications*. Continuous Multivariate Distributions. Wiley. ISBN: 9780471654032. URL: <https://books.google.ch/books?id=EbPBXJ-N-m4C>.
- Krzanowski, W. J. (1993): "The location model for mixtures of categorical and continuous variables". In: *Journal of Classification* 10.1, pp. 25–49. ISSN: 0176-4268. DOI: [10.1007/bf02638452](https://doi.org/10.1007/bf02638452).
- Lettry, Loyal Christine (2021): *rrclust: Clustering observations from the Swiss Old-Age and Survivors's Insurance (OASI) Register of Rents*. URL: <https://github.com/asam-group/rrclust>.
- Milligan, Glenn W. (1980): "An examination of the effect of six types of error perturbation on fifteen clustering algorithms". In: *Psychometrika* 45.3, pp. 325–342. ISSN: 0033-3123. DOI: [10.1007/bf02293907](https://doi.org/10.1007/bf02293907).
- Modha, Dharmendra S. and W. Scott Spangler (2003): "Feature Weighting in k-Means Clustering". In: *Machine Learning* 52.3, pp. 217–237. ISSN: 0885-6125. DOI: [10.1023/a:1024016609528](https://doi.org/10.1023/a:1024016609528).
- Olkin, I. and R. F. Tate (1961): "Multivariate Correlation Models with Mixed Discrete and Continuous Variables". In: *The Annals of Mathematical Statistics* 32.2, pp. 448–465. ISSN: 0003-4851. DOI: [10.1214/aoms/1177705052](https://doi.org/10.1214/aoms/1177705052).
- Tibshirani, Robert and Guenther Walther (2005): "Cluster Validation by Prediction Strength". In: *Journal of Computational and Graphical Statistics* 14.3, pp. 511–528. ISSN: 10618600. URL: <http://www.jstor.org/stable/27594130>.
- Vermunt, Jeroen and Jay Magidson (June 2002): "'Latent class cluster analysis,' in Applied Latent Class Analysis, eds J". In: *A. Hagenaars and A. L. McCutcheon*, pp. 89–106. DOI: [10.1017/CB09780511499531.004](https://doi.org/10.1017/CB09780511499531.004).
- Wickham, Hadley et al. (2021): *dplyr: A Grammar of Data Manipulation*. URL: <https://CRAN.R-project.org/package=dplyr>.
- Wolfe, Jason, Aria Haghighi, and Dan Klein (2008): "Fully Distributed EM for Very Large Datasets". In: *Proceedings of the 25th International Conference on Machine Learning*. ICML '08. Helsinki, Finland: Association for Computing Machinery, pp. 1184–1191. ISBN: 9781605582054. DOI: [10.1145/1390156.1390305](https://doi.org/10.1145/1390156.1390305). URL: <https://doi.org/10.1145/1390156.1390305>.

Authors

Layal Christine LETTRY

Department of Informatics, University of Fribourg, 1700 Fribourg, Switzerland

Bd de Pérolles 90, 1700 Fribourg, Switzerland, E-mail: layal.lettry@gmail.com, Web: layalchristinelettry.rbind.io

Abstract

The anonymous data of the Swiss Pension Register (CCO/FSIO) (PR) are typically used to estimate (in the short, middle and long term) the future revenues and expenditures of the Old-Age and Survivors' Insurance (OASI). In this perspective, it is essential to have a clear look at the register's main statistical features. To better understand it and benefit more from its richness, we propose analysing the raw data by an appropriate clustering method.

Jel Classification

JEL: C38

Keywords

Kamila; Clustering; R; AVS; AHV; OASI; Swiss Pension Register; FSIO; prediction strength criterion; classification; RAMD; AADR; UniFr

Working Papers SES collection

Last published

520 Eugster N., Ducret R., Isakov D., Weisskopf J-P.: Chasing dividends during the COVID-19 pandemic; 2020

521 Loginova D., Portmann M. and Huber M. Assessing the effects of seasonal tariff-rate quotas on vegetable prices in Switzerland; 2020

522 Herz H, Zihlmann C.: Adverse Effects of Monitoring: Evidence from a field experiment; 2021

523 Grossmann V.: Das House Kapital; 2021

524 Ducret R.: Investors' perception of business group membership during an economic crisis. Evidence from the COVID-19 pandemic; 2021

525 Dubois C., Lambertini L., Wu Yu: Gender Effects of the Covid-19 Pandemic in the Swiss Labor Market; 2022

526 Herzy H., Kistlerz D., Zehnder C., Zihlmann C., Hindsight Bias and Trust in Government: Evidence from the United States; 2022

527 De Chiara A., Engl F., Herz H., Manna E., Control Aversion in Hierarchies; 2022

528 Buechel B., Klößner S., Meng F., Nassar A.: Misinformation due to asymmetric information sharing; 2022

Catalogue and download links

<http://www.unifr.ch/ses/wp>

http://doc.rero.ch/collection/WORKING_PAPERS_SES

Publisher

Université de Fribourg, Suisse, Faculté des sciences économiques et sociales et du management Universität Freiburg, Schweiz, Wirtschafts- und sozialwissenschaftliche Fakultät University of Fribourg, Switzerland, Faculty of Management, Economics and Social Sciences

Bd de Pérolles 90
CH-1700 Fribourg
Tél.: +41 (0) 26 300 82 00
decanat-ses@unifr.ch
www.unifr.ch/ses