Fabian Peter Pribahsnik, BSc.

Let the data tell us their story
Machine learning with insurance policies

Master's Thesis

to achieve the university degree of

Master of Science

Financial and Actuarial Mathematics

submitted to

Vienna University of Technology

Institute of Statistics and Mathematical Methods in Economics

Supervisor

Univ.Prof. Dipl.-Math. Dr.rer.nat. Thorsten Rheinländer

Vienna, October 2017

Affidavit

I declare that I have authored this thesis independently, that I have not used other than the declared sources/resources, and that I have explicitly indicated all material which has been quoted either literally or by content from the sources used.

Date	Signature

Abstract

Insurance companies are facing a huge amount of regulations, including various guidelines addressing forecast scenario calculations for the policies in the portfolio. Taking the hundreds of thousands policies into account an average insurance company has in its portfolio on can easily see that these scenario calculations are very time consuming. Due to the rising number of policies and the very tight time schedule introduced with Solvency II insurance companies are looking for ways to reduce the computational time significantly. In the past years different approaches were developed and already used for grouping similar policies together and therefore reducing the computation time. The currently used algorithms are ranging from just grouping policies with exactly the same attributes together to basic cluster algorithms like k-means. This work highlights potential problems with the algorithms currently used and tries to implement some machine learning techniques to accomplish the task of grouping.

Contents

Abstract **v**

1. Introduction **1**
1.1. Legal Framework 2
1.2. Statistical Learning 4

2. Sensitivities **7**
2.1. Age 9
2.2. Technical interest rate 11
2.3. Duration 14

3. k-means **19**
3.1. Elbow method - gap statistic 26
3.2. Silhouette method 28
3.3. Curse of dimensionality 32

4. Non-negative least squares (NNLS) **35**

A. Tables **37**

List of Figures

2.1. Logarithm of the yearly mortality defined by the Austrian Annuity Valuation Table AVÖ 2005R Unisex. 9
2.2. Yearly cash flow for claims depending on the age and the technical interest rate. 12
2.3. Premiums depending on the age and the technical interest rate. 12
2.4. Present value of future profits at time 0 depending on the age and the technical interest rate. 13
2.5. Yearly reserve depending on the age and the technical interest rate. 13
2.6. Yearly cash flow for claims depending on the duration and the age. 16
2.7. Premiums depending on the duration and the age 16
2.8. Present value of future profits at time 0 depending on the duration and the age. 17
2.9. Yearly reserve depending on the duration and the age. 17

3.1. Results for a three-cluster example: (a) raw data; (b) $k = 2$; (c) $k = 3$; (d) $k = 5$ 25
3.2. Three-cluster example: (a) Total within Sum of Squares; (b) Gap-Statistic 27
3.3. Left: silhouette plot for $k = 3$; Right: silhouette plot for $k = 5$. 31

Todo list

Weitere Details für jedes Kapitel folgen. 2
Make a cake 39
Figure: Make a sketch of the structure of a trebuchet. 39

1. Introduction

A fundamental problem faced by many insurance companies is the projection of the insurance portfolio into the future. In a projection, the future cash flows for each individual policy must be calculated on the basis of actuarial principles and then saved for further analysis. Depending on the purpose of the forecast, these cash flows have to be provided either on a monthly or annually basis.Together with the contractually agreed benefits to the policyholder, a wide variety of other parameters must also be taken into account and modelled in such projections. These include, for example, all contract changes that may occur during the duration of the contract, such as a premium pause, a surrender or the occurrence of a claim. While the calculation of a small set of policies over a relatively short time horizon does not pose a challenge in terms of computation time, this fact changes greatly when projecting entire portfolios. This problem is particularly severe for life insurance portfolios. These contracts often have durations of several decades and the portfolios tend to grow over time as fewer policyholders leave than new ones are added. Even with the optimistic assumption that the complete projection of a single contract only takes a hundredth of a second, the computing time adds up to several hours for a portfolio with several million policies. As the liabilities are projected over a period of several decades, it is also essential to simulate the development of the assets, underlying the liabilities, over this period. Even under the assumption that the assets can be represented by a small number of different types of investments, their projection additionally increases the runtime of the projection. The simultaneous simulation of the asset and liability portfolios results in further factors that have to be taken into account with respect to the behaviour of policyholders during the contract term. If one assumes that the policyholder behaviour during the projection period also depends on external parameters such as the current interest rates on savings, one also has to include that effect into the simulation. This finally results in a dynamic interaction of all components which can be summarized as follows:

- Project all contracts to the next year

- Simulate the development of the assets
- Determine how to policyholders will behave and whats the effect on the assets
- Simulate contracts again

This thesis reviews the currently used technique for grouping policies together and introduces furthermore some new approaches on how life insurance policies can be grouped together. We will therefore highlight drawbacks and advantages with a special emphasis on the regulatory requirements of every approach discussed. Theoretical considerations as well as practical implementations and tests with real world data will provide us some information on which method an insurance company should work with in order to obtain the best grouping results.

This thesis is structured as follows: First we give an overview on the legal framework which lays down the minimum requirements for grouping-approaches in insurance companies and introduce the two types of statistical learning, namely supervised and unsupervised. We then discuss how sensitive main characteristics of a policy are with respect to the interest rate, the age or the duration to get a better understanding on which parameters are important for grouping purposes. We therefore make a sensitivity analysis with real world data and a widely used projection tool . In the next chapter we introduce the currently used unsupervised learning algorithm k-means and derive some theoretical findings.

Weitere Details für jedes Kapitel folgen.

1.1. Legal Framework

Solvency II - entered into force on 1 January 2016 - is the European framework for a common insurance supervision. It is intended to achieve a harmonization of the European insurance sector and was implemented in accordance with the Lamfalussy architecture which works on a 4 level basis [**Lamfalussy˙homepage**]. The most significant elements and aims of the new regulation framework can be studied on the homepage of the financial market authority (FMA) [**FMA˙homepage**] and on the homepage of the European Insurance and Occupational Pensions Authority (EIOPA) [**EIOPA˙homepage**]. This work is intended not to cover all aspects and aims of the new Solvency II regulation framework but focuses on the topic of data quality regarding to

the actuarial function. In order to meet all the requirements imposed by Solvency II, insurance companies need to process large amounts of data within a short period. One critical aspect of these calculations is the projection horizon which however should cover the full lifetime of all obligations as stated in [**Time˙horizon**]:

3.83.

> The projection horizon used in the calculation of best estimate should cover the full lifetime of all obligations related to existing insurance and reinsurance contracts on the date of the valuation.

3.84.

> The determination of the lifetime of insurance and reinsurance obligations shall be based on up-to-date and credible information and realistic assumptions about when the existing insurance and reinsurance obligations will be discharged or cancelled or expired.

Another aspect needed to be considered is the fact that cash flow calculations need to be done for a variety of different economic scenarios which yields to an enormous computational effort. Due to the tight time schedule, insurance companies are looking for new possibilities to speed up these time consuming calculations. One approach is not to make all these calculations on a per policy level, but on a grouped level where similar policies are grouped together and represented by only a few policies. This approach raises the question of how to maintain data quality as mentioned in the level 1 directive [**Directive**] while reducing the number of policies.

Article 82

Data quality and application of approximations, including case-by-case approaches, for technical provisions

> Member States shall ensure that insurance and reinsurance undertakings have internal processes and procedures in place to ensure the appropriateness, completeness and accuracy of the data used in the calculation of their technical provisions...

By publishing the level 2 regulations, supplementing the level 1 directive [**Directive**] the European Commission is getting more specific on data quality (Article 19 in [**Regulations**]) and also formulates concrete requirements for grouped policies [**Regulations**].

Article 35

Homogeneous risk groups of life insurance obligations

The cash flow projections used in the calculation of best estimates for life insurance obligations shall be made separately for each policy. Where the separate calculation for each policy would be an undue burden on the insurance or reinsurance undertaking, it may carry out the projection by grouping policies, provided that the grouping complies with all of the following requirements:

a) there are no significant differences in the nature and complexity of the risks underlying the policies that belong to the same group;

b) the grouping of policies does not misrepresent the risk underlying the policies and does not misstate their expenses;

c) the grouping of policies is likely to give approximately the same results for the best estimate calculation as a calculation on a per policy basis, in particular in relation to financial guarantees and contractual options included in the policies.

These level 2 regulations are a reference point on what to consider when grouping policies together and they are even further specified in the level 3 guidelines issued by EIOPA[**Guidelines˙TP**]. Further details on the level 3 guidelines including feedback statements to the consultation paper (EIOPACP-14/036) and the guidelines can be obtained from [**Final˙Report**].

1.2. Statistical Learning

Statistical learning refers to a set of methods which deals with predicting outcomes based on input variables or finding patterns in data sets. In order to accomplish the task of grouping together similar policies, different approaches from statistical learning can be applied. All these methods can be classified either as supervised or unsupervised. Within the framework of supervised methods, statistical models try to predict output variables y_i based on some input variables x_i where the relation $y = f(x)$ is unknown. It is therefore indispensable to have input as well as output data to parameterize such a model in order to find a prediction $\hat{f}$ of f. Unsupervised methods, in contrast, are used when inputs x_i but no corresponding outputs y_i are available. These

methods then try to find some hidden patterns within to data. The task of grouping insurance policies involves many different aspects. On the one hand we have all data needed to apply supervised methods, but on the other hand we are only interested in the patterns that can be revealed by an unsupervised method. The input variables are given by the characteristics of each policy and the corresponding output variables are determined by the projection tool. Our primary goal is not to get a good $\hat{f}$ because the projection tool, which stands for f, is already known. We are more interested in hidden patterns that can be used for grouping purposes. In a first step we will apply unsupervised methods to the data and try to group the policies based on their characteristics and their cash flows. In a further step we will try to use the additional information of f to improve the grouping results if possible.

2. Sensitivities

Grouping together single policies and representing them by just a few representative ones always comes along with a loss of information. A natural question which arises when grouping insurance contracts together is how to determine the main characteristics of the new representative policy. Some characteristics should for technical reasons be defined as the sum of the individual ones, like the sum insured, the premium or the accumulated reserve. This is needed to guarantee the equality between the un-grouped and the grouped portfolio in terms of these characteristics at the beginning of the projection horizon. For other characteristics like the age, the duration or the gender it is not intuitively clear how they should be defined for a representative policy. Possible solutions which can be implemented easily range from taking the weighted average over the value with the highest relative frequency or to just taking the median of the grouped policies. Another difficulty which arises when grouping together policies from different product generations is, how the technical interest rate of the representative contract should be defined. Even if the policies are identical in terms of age, sex, sum insured, duration, costs,... and just differ on their issue date the huge possible differences with respect to the technical interest rate as shown in table 2.1 can have enormous impacts on the projected cash flows. Already a relative small difference in the technical interest of only one percent causes double digit differences in the guaranteed capital after 1 decade.

31.12. 1994	30.06. 2000	31.12. 2003	31.12. 2005	31.03. 2011	20.12. 2012	31.12. 2014	31.12. 2015	31.12. 2016
4%	3.25%	2.75%	2.25%	2%	1.75%	1.5%	1%	0.5%

Table 2.1.: Maximum technical interest rates for life insurance contracts issued after the given dates. (see [**fakten˙trends**])

It is therefore important to know how sensitive the different output variables of interest which are calculated by the projection tool react if various input parameters are changed slightly. The most basic task is to determine whether the correlation between the input and output variables is positive or negative.

2. Sensitivities

There are many output variables which are important for determining if a grouping process has been successful in terms of accuracy or not, but in the subsequent we will focus only on a few of them, namely the premium, the present value of future profits at time 0, the reserve and the yearly claims. Due to the big variety of different insurance products we will just give some general guidelines based on the most important input variables. Most of the life insurance contracts can be built up by the following elementary insurance types and some additional factors for different types of costs. [1]:

$$A_x = \sum_{k=0}^{\infty} v^{k+1}{}_k p_x q_{x+k} \qquad \text{(Whole life insurance)} \tag{2.1}$$

$$A^1_{x:\overline{n}|} = \sum_{k=0}^{n-1} v^{k+1}{}_k p_x q_{x+k} \qquad \text{(Term insurance)} \tag{2.2}$$

$$A_{x:\overline{n}|} = \sum_{k=0}^{n-1} v^{k+1}{}_k p_x q_{x+k} + v^n{}_n p_x \qquad \text{(Endowment)} \tag{2.3}$$

$$\ddot{a}_x = \sum_{k=0}^{\infty} v^k{}_k p_x \qquad \text{(Whole life annuity)} \tag{2.4}$$

$$\ddot{a}_{x:\overline{n}|} = \sum_{k=0}^{n-1} \ddot{a}_{\overline{k+1}|} {}_k p_x q_{x+k} + \ddot{a}_{\overline{n}|}\ {}_n p_x \qquad \text{(Temporary life annuity)} \tag{2.5}$$

These basic types already show that the main characteristics which should be taken care of, when it comes to a grouping process are the age x, the duration n and the technical interest rate which is implicitly given by v. The following graphs and analyses are based on an endowment policy with a duration of 25 years, a sum insured of 10.000 € and an investment return of 3% p.a. over the whole projection horizon. The duration of 25 years is in all subsequent considerations equal to the duration of the premium payments which are made on a yearly basis. All other parameters especially the lapse, paid-up and surrender rates as well as first and second order assumptions can't be given here in detail.

[1] For detailed definitions and explanations see [**Gerber**].

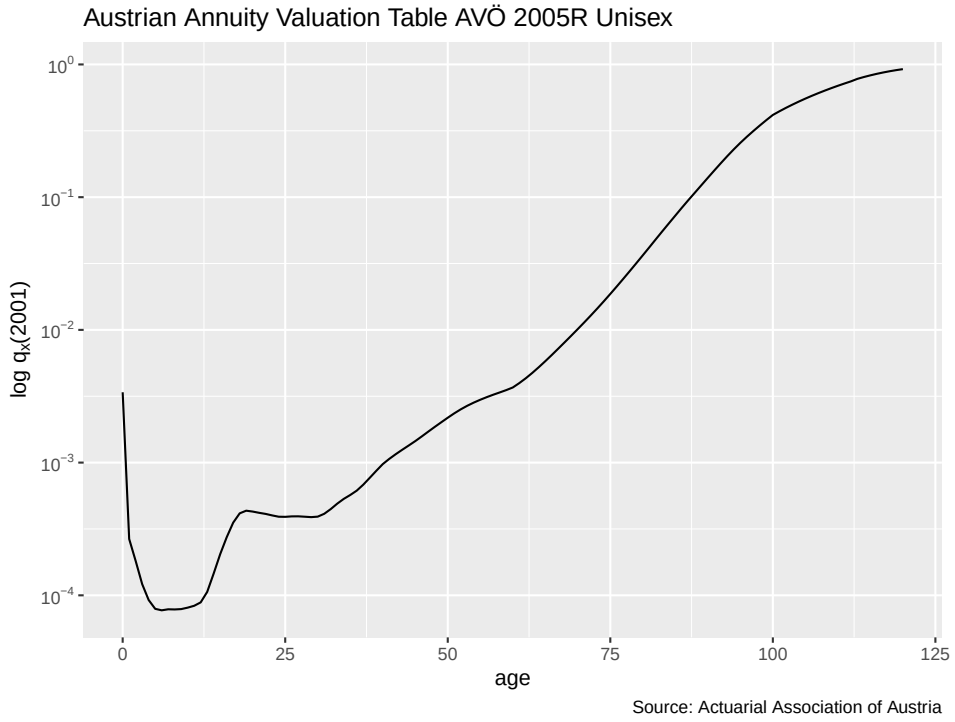


Figure 2.1.: Logarithm of the yearly mortality defined by the Austrian Annuity Valuation Table AVÖ 2005R Unisex.

2.1. Age

The age of an insured person is one of the main factors which drives the projected outcome because it directly influences the probability of death and survival as shown in (2.1) - (2.5). When the age is changed from x to $x+1$ the survival- and death-probabilities ${}_kp_x$ and ${}_kq_x$ change as well. It is impossible to predict in general whether the probabilities will rise or fall. Figure (2.1) shows, for example, the graph of logarithmic mortality rates based on the values of the unisex mortality table from the Actuarial Association of Austria [**kainhofer2006new**]. A high level of non-linearity can be observed, which naturally leads to greater challenges in the grouping.

As known from life tables it is a bit more likely to die just after birth than a bit afterwards and the same is true for people aged around 20. The exact ages where the probability of survival increases and the probability of death decreases when a person gets a year older depends heavily on the life table and the sex of the insured person. Whether the values for (2.1) - (2.5) will rise or

fall when the age x is increased by 1 year will therefore depend on n, x and the sex of the insured person. To get a better insight into the portfolio, simulation runs for various parameters should be done. In figure (2.2) the development of the yearly total claims is plotted against the duration of 25 years. The claims are the amount of money that must be paid to the policyholder at the time of an insured event multiplied by the probability that such an event will occur. Such events can be of all kinds, but the most common are, for example, the death of the insured person or a surrender of the contract.

For a clearer chart the last cash flow which is the sum insured and therefore substantially larger than the yearly claims is omitted. We see that for younger people (green and red lines) the claims are almost identical in the first years and only deviate slightly at the end of the duration due to minor differences in the probabilities of death. For an insured person aged 60 we observe over the entire projection horizon considerably higher claims compared to younger policyholders. This gap between young and old policyholders which is mostly driven by mortality effects even increases with time. An insured person aged 60 which gets one year older faces in absolute values an higher increase in the mortality rate compared to an insured person aged 40 and so the claims will be higher in absolute values for the older person. This effect is partially compensated by lower surrender claims due to the higher mortality rates. If one compares the development of the claims also with respect to different interest rates, one can see in figure (2.2) that there are hardly any differences between the values of 2%, 3% and 4% shown.

In figure (2.3) the development of the booked premium at time 0 is plotted against the age. For policyholders aged between 15 and 30 the premium stays almost constant and then starts to increase exponentially. The increase of the premium is of exponential order due to the fact that the mortality rate is also increasing exponentially. Another fact that is not surprising is that the premium is the lower the higher the technical interest rate is, because the technical interest rate is used as a discount factor in (2.1) - (2.5). In figure (2.4) the present value of future profits (PVFP) at time 0 is plotted against the age. The PVFP is the higher the higher the entry age of the policyholder is, which is a counter intuitive relation at first sight. An analysis of the yearly cashflows for two different policyholders aged 15 and 70 reveals that this phenomenon is based on two different aspects as given in detail in table A.1.

1. When the guaranteed interest rate is roughly equal to the investment return or even higher then it is more advantageous for the insurance company when the policyholder is older and therefore dies earlier because

the difference between the guaranteed interest rate and the investment return need not to be financed over a long period. This leads to the observed fact that the PVFP is the lower the higher the technical interest rate is.

2. The differences of the first and second order assumptions of the mortality rates are in absolute values the bigger the higher the age is, because in almost all cases the second order assumptions of of the mortality rates are just a fixed fraction of the first order assumptions. This directly leads to a higher risk margin and therefore to a higher premium for elder persons in absolute values as shown in column *prem_diff* in table A.1. The higher premium overcompensates the higher death claims and therefore increases the surplus in absolute values and leads to a higher present value of future profits.

In figure (2.5) the value of the reserve is plotted against the time. In all three cases the reserve is zero at the beginning and then starts to increase. This is due to the fact, that the valuation date of the projection is Q1 but the begin month of the policy is later. We see that for the ages of 15 and 30 the difference in the reserve is negligible for all different values of the technical interest rate. The reserve for a 45 year old person is almost identical to the one of younger policyholders at the first 10 years of the endowment but then increases at a slightly slower rate. For a person aged 60 we get a different picture because the reserve is not always monotonically increasing as it is true for the younger policyholders. The reserve increases after approximately 10 years at a much slower rate compared to the other policyholders and even starts to decrease after roughly 20 years. This effect can again be explained by the much higher morality rates in absolute values for older policyholders as time increases. Due to that fact the reserve is at the end of the projection horizon when the maturity claims are payed out for an older policyholder approximately half of the size as for a younger policyholder.

2.2. Technical interest rate

The technical interest rate is one of the key assumptions in the life insurance business. It determines the factor by which the reserve and the savings premium increases during the contract period. After the contract has been concluded the technical interest rate is fixed and can't be changed by the insurance company. The maximum technical interest rate has been reduced

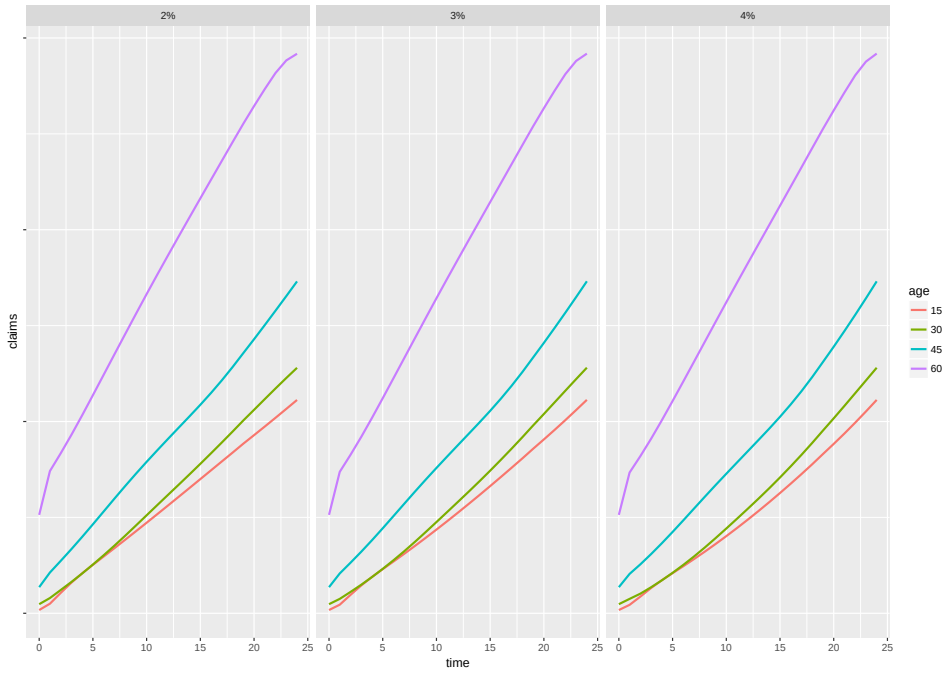


Figure 2.2.: Yearly cash flow for claims depending on the age and the technical interest rate.

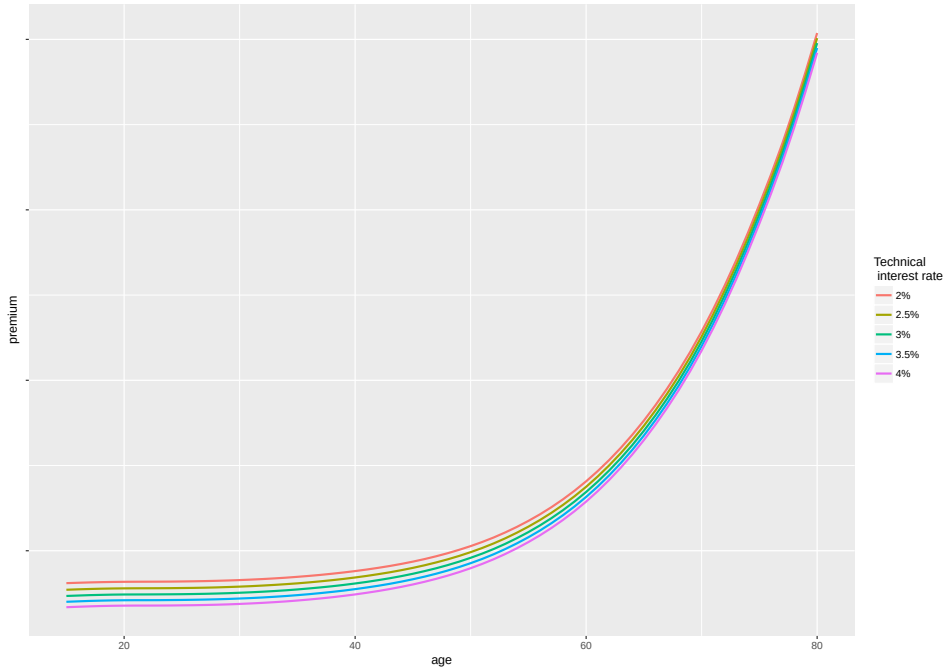


Figure 2.3.: Premiums depending on the age and the technical interest rate.

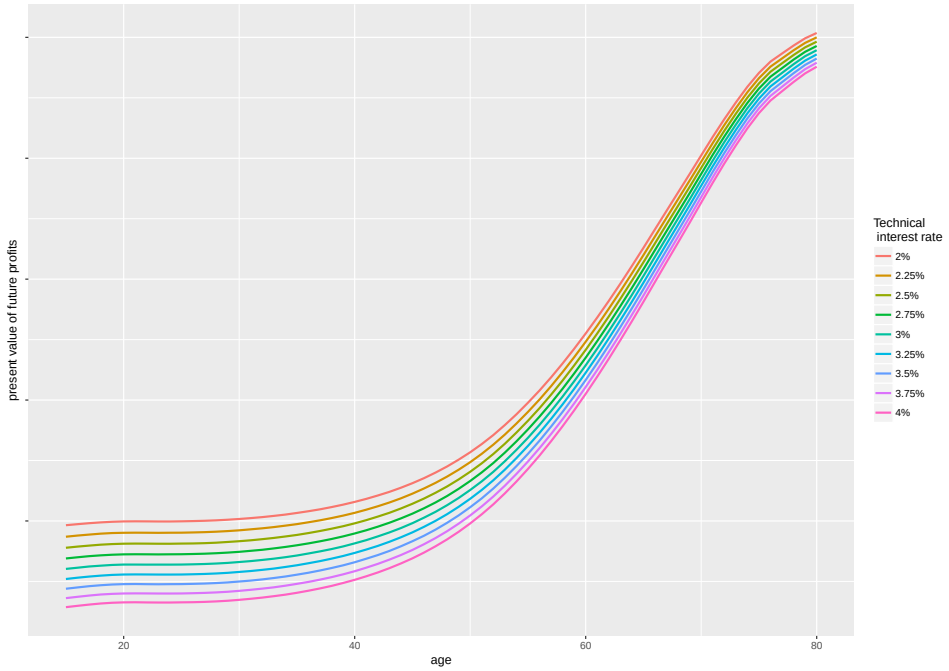


Figure 2.4.: Present value of future profits at time 0 depending on the age and the technical interest rate.

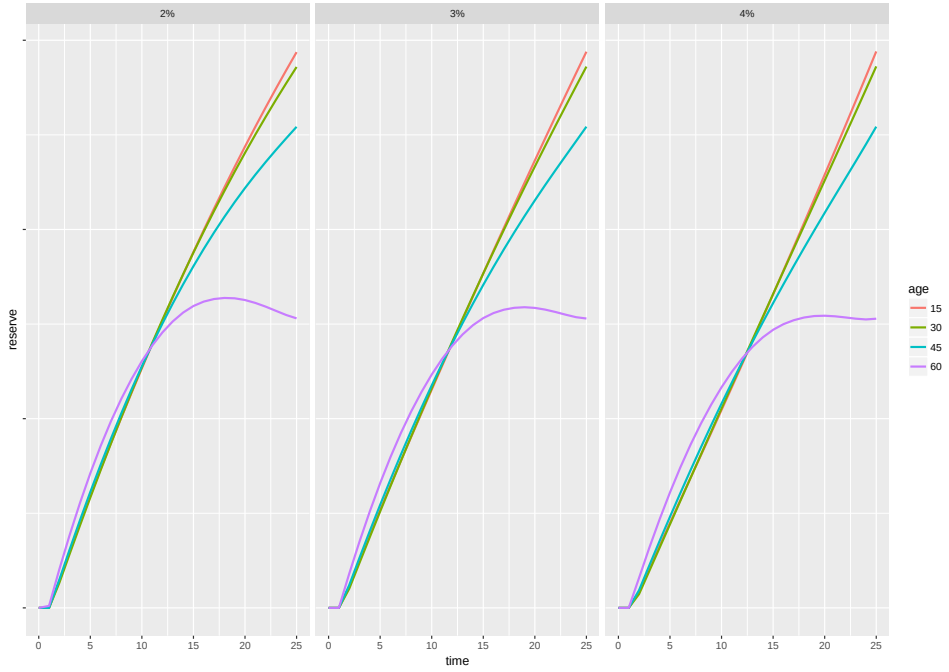


Figure 2.5.: Yearly reserve depending on the age and the technical interest rate.

dramatically by the Financial Market Authority in recent years from 4% to 0.5% as shown in table 2.1. The regulation of this upper bound is also relevant as it is often related to the minimum interest rate guaranteed to the policy holder. It is obvious that the higher the technical interest rate, the higher the guaranteed benefit or the lower the premium will be. When the grouping algorithm forces policies from different product generations with same payout characteristics but different technical interest rates to be grouped together one important thing to be aware of are sensitivities. Another important aspect which need to be taken care of when policies with different technical interest rates are grouped together is the one of consistent management rules. Take for example policy 1 with a technical interest rate of 2% and a bonus rate of 2% and policy 2 with a technical interest rate of 4% and a bonus rate of 0%. Lets assume that the two policies are similar and the grouped policy has a technical interest rate of 3% and a bonus rate of 1%. The total interest rate then is the same for the grouped and the ungrouped policies. Assume that the bonus rate is reduced by 1% caused by a management decision. Then policy 1 has only a bonus rate of 1% and policy 2 doesn't change at all, but the grouped policy has now a bonus rate of 0%. The total interest rate is not the same for the grouped and the ungrouped policies which can potentially have major impacts on the projected cash flows. When the technical interest rate increases, the present value of future profits as well as the premium will decrease as shown in figure (2.4) and (2.3) respectively. This behavior is not surprising at all, because as the technical interest rate rises the insurance company guarantees a higher benefit and this yields ceteris paribus to a lower PVFP. The reserve and the claims are not that much affected by an increase of the technical interest rate as shown in figure (2.2) and (2.5) respectively.

2.3. Duration

The duration n of an insurance contract is next to the age and the technical interest rate another main characteristic which need to be taken care of when a grouping process is carried out. In figure (2.6) the yearly claims except the last claim, which is the maturity claim, are shown for different ages. One obvious conclusion that can be derived is that the sum of all sorts of claims except the maturity claim is getting the higher the higher the age is. Another observation that can be made is that for any given time t the claims are the higher the shorter the duration gets when we keep the age fixed. This is not surprising at all, because a shorter duration goes along with a higher premium (see figure

(2.7)) and a higher reserve (see figure (2.9)) which yields to higher claims for every fixed t. In figure (2.7) we see that the premium is decreasing with an exponential order when the duration is increased. The difference between the premiums for policyholders with different ages is indistinguishable small for short term contracts and is getting bigger as duration increases. In figure (2.8) the present value of future profits at time 0 is plotted against the duration of the contract. We see the same effect as in figure (2.4) where contracts with higher ages lead to a higher PVFP. The effect of the absolute difference between the first and second order mortality assumptions is getting the bigger the longer the duration is and therefore the PVFP is getting the higher the longer the duration is. In the last sensitivity chart (2.9) the reserve is plotted against the time for different values of x and n. We see that for every time t the reserve is the higher the shorter the duration is, because a shorter duration leads, ceteris paribus, to a higher premium which then results in a higher reserve. For short term contracts up to 10 years the reserve is approximately the same across different ages and is then getting the lower the higher the age and the longer the duration is.

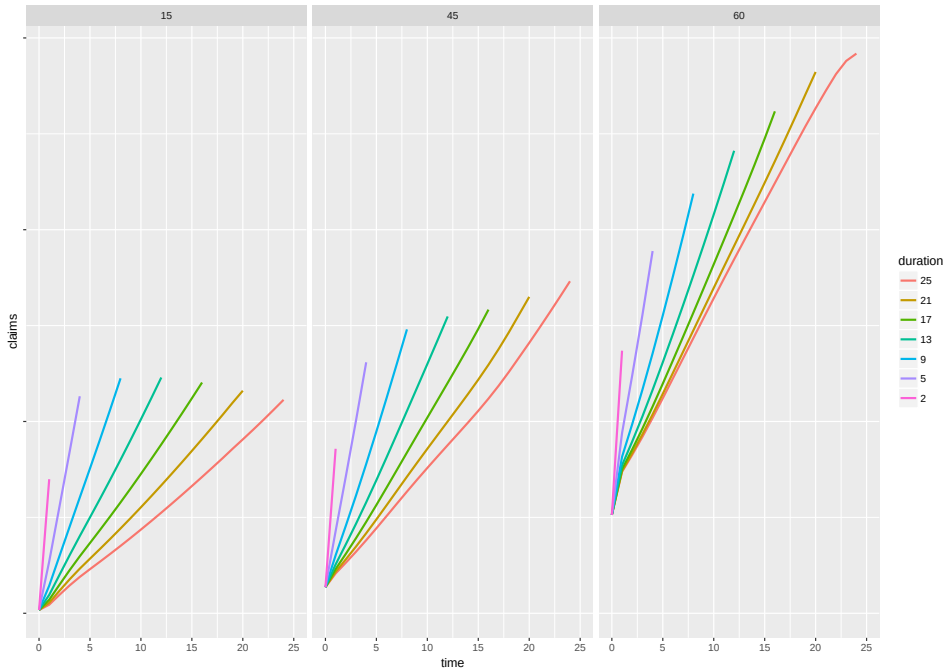


Figure 2.6.: Yearly cash flow for claims depending on the duration and the age.

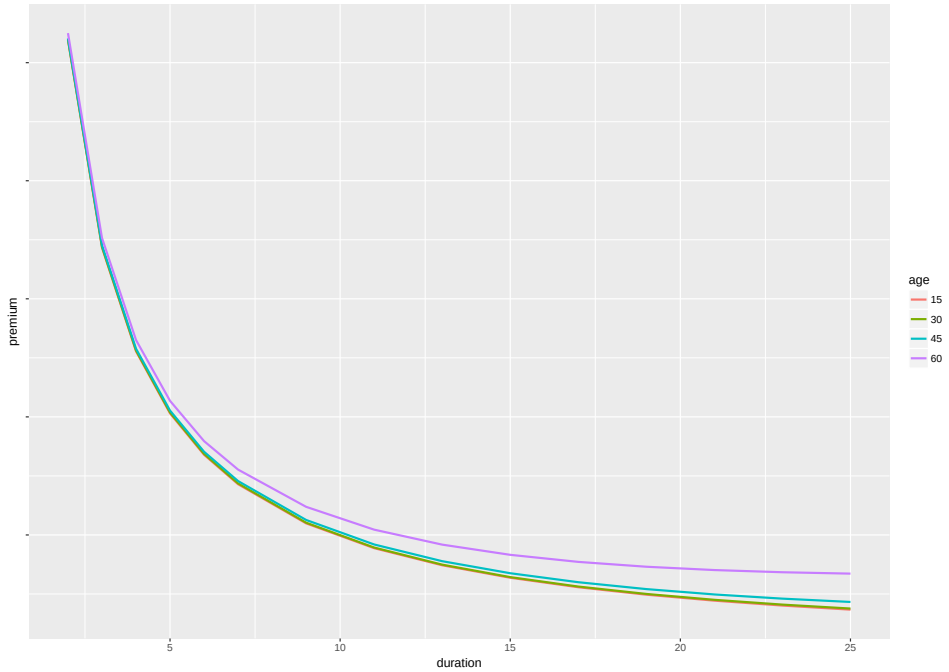


Figure 2.7.: Premiums depending on the duration and the age

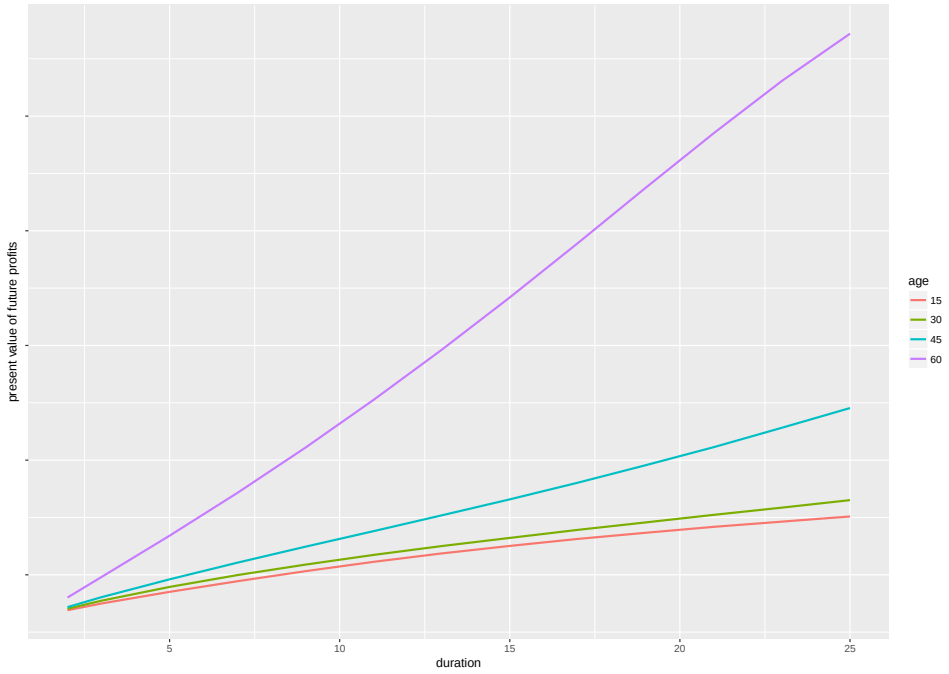


Figure 2.8.: Present value of future profits at time 0 depending on the duration and the age.

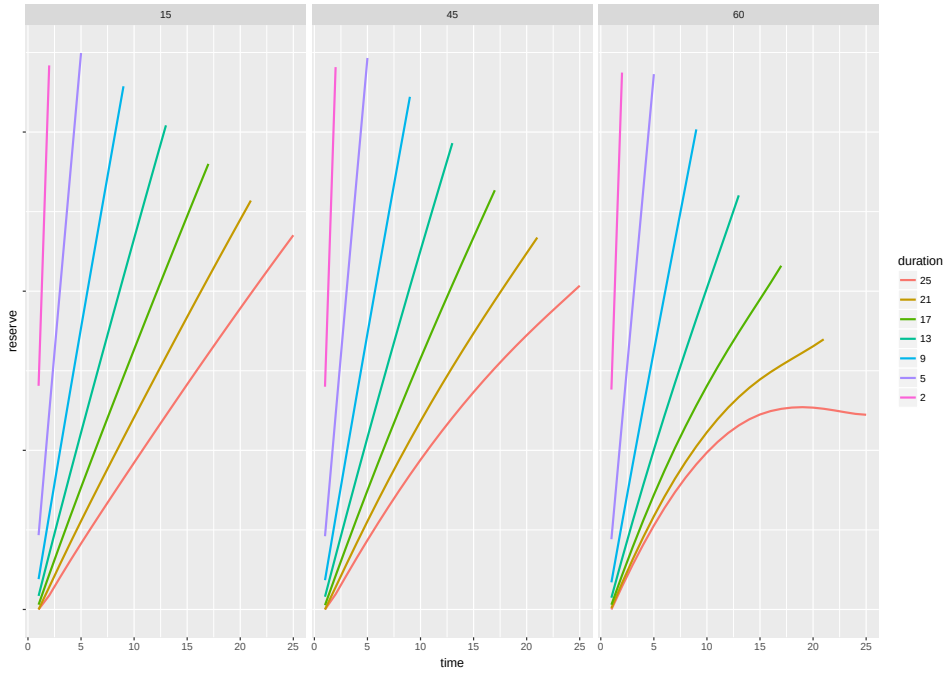


Figure 2.9.: Yearly reserve depending on the duration and the age.

3. k-means

To be able to process, summarize and understand huge amounts of data better, one is interested in methods that are able to find patterns in the data. The characteristics of these patterns then can be represented by just a few representative data points which behave like the whole data set. Given a data set the challenge is, based on a measure of similarity, to find groups of observations which are quite similar within each group but quite different to all the other groups. If this task has to be done with unlabeled data it is referred to as unsupervised clustering (c.f.[**jain2010data**]). One of the most widely used unsupervised clustering approaches is the k-means clustering. The k-means method is a simple approach in cluster analysis which splits a data set of n p-dimensional observations into k distinct clusters. Each observation belongs uniquely to exactly one of the k clusters, where k is a predefined number of clusters. Let $C = \{C_1, C_2 \ldots, C_k\}$ denote the sets containing the indices of the observations related to the clusters, then we get:

Definition 3.1. *Let $X = \{x_1, ..., x_n\}$ be a data set. X is said to be partitioned into k different clusters $C_1, C_2 \ldots, C_k$ if*

$$\textit{(i)}\ C = C_1 \cup C_2 \cdots \cup C_k = \{1, \ldots, n\}$$

$$\textit{(ii)}\ C_i \cap C_j = \emptyset \quad \forall i \neq j$$

The basic idea of the k-means clustering is to minimize the variation within the clusters. For this purpose some distance measure is needed in order to be able to define variation within clusters.

Definition 3.2. *Let X be a set, d: $X \times X \to \mathbb{R}$ a function. Then d is called a metric (or distance) on X if for all x, y, $z \in X$ the following conditions are fulfilled:*

(D_1) $d(x, y) = 0 \Leftrightarrow x = y$ *(Identity of indiscernibles)*

(D_2) $d(x, y) = d(y, x)$ *(symmetry)*

(D_3) $d(x, z) \leq d(x, y) + d(y, z)$ *(triangle inequality)*

Remark 3.1. *Given the axioms from definition 3.2 it can be shown that a non-necgativity property can be deducted e.g.* $d(x,y) \geq 0 \ \forall x, y \in X$.

$$\begin{aligned} d(x,y) + d(y,x) &\geq d(x,x) \\ d(x,y) + d(x,y) &\geq d(x,x) \\ 2d(x,y) &\geq 0 \\ d(x,y) &\geq 0 \end{aligned}$$

Example 3.1. *(c.f. [***analysis˙1***]) The most common used distance functions for two points* $x = (x_1, ..., x_p)$ *and* $y = (y_1, ..., y_p)$ *in* $\mathbb{R}^p$ *are:*

- ⋆ *Euclidean distance:*

$$d_2(x,y) := \sqrt{\sum_{j=1}^{p} (x_j - y_j)^2}$$

- ⋆ *Manhattan distance:*

$$d_1(x,y) := \sum_{j=1}^{p} |x_j - y_j|$$

- ⋆ *Chebyshev (maximum) distance:*

$$d_\infty(x,y) := \max_j |x_j - y_j|$$

- ⋆ *Minkowski distance (*L^q *distance) with* $q \geq 1$*:*

$$d_p(x,y) := \left(\sum_{j=1}^{p} (x_j - y_j)^q \right)^{\frac{1}{q}}$$

Typically the Euclidean distance is used as a measure of similarity to compute the distance between the different points. By using the Euclidean metric as a measure of similarity one assumes that the clusters are spherical.

Definition 3.3. *Let the Euclidean metric be the measure of similarity for the data points in the data set* $X = \{x_1, ..., x_n\}$ *and* p *the dimension of the data. Then the variation within one cluster* C_l, $l = 1, ..., k$ *is defined as:*

$$D(C_l) := \frac{1}{|C_l|} \sum_{j=1}^{p} \sum_{i,i' \in C_l} (x_{ij} - x_{i'j})^2$$

Remark 3.2. *For the average over one dimension in one cluster we use the short notation* $\bar{x}_{lj} = \frac{1}{|C_l|} \sum_{i \in C_l} x_{ij}$.

Remark 3.3. *The identity* $\sum_{i \in C_l} (x_{ij} - \bar{x}_{lj})^2 = \left(\sum_{i \in C_l} x_{ij}^2 \right) - |C_l| \bar{x}_{lj}^2$ *can be verified by simple calculus.*

Corollary 3.1. *The variation within one cluster,* $D(C_l)$ *can be written as:*

$$D(C_l) = 2 \sum_{i \in C_l} \sum_{j=1}^{p} (x_{ij} - \bar{x}_{lj})^2 \tag{3.1}$$

Proof.

$$\begin{aligned}
D(C_l) &= \frac{1}{|C_l|} \sum_{j=1}^{p} \sum_{i,i' \in C_l} (x_{ij} - x_{i'j})^2 \\
&= \frac{1}{|C_l|} \sum_{j=1}^{p} \sum_{i,i' \in C_l} x_{ij}^2 - 2 x_{ij} x_{i'j} + x_{i'j}^2 \\
&= \sum_{j=1}^{p} \left(\frac{1}{|C_l|} \sum_{i,i' \in C_l} x_{ij}^2 - 2 \frac{1}{|C_l|} \sum_{i,i' \in C_l} x_{ij} x_{i'j} + \frac{1}{|C_l|} \sum_{i,i' \in C_l} x_{i'j}^2 \right) \\
&\overset{(Remark\ 3.2)}{=} \sum_{j=1}^{p} \left(\sum_{i \in C_l} x_{ij}^2 - 2 \bar{x}_{lj} \sum_{i \in C_l} x_{ij} + \sum_{i' \in C_l} x_{i'j}^2 \right) \\
&\overset{(Remark\ 3.2)}{=} 2 \sum_{j=1}^{p} \left(\sum_{i \in C_l} x_{ij}^2 - |C_l| \bar{x}_{lj}^2 \right) \\
&\overset{(Remark\ 3.3)}{=} 2 \sum_{i \in C_l} \sum_{j=1}^{p} (x_{ij} - \bar{x}_{lj})^2
\end{aligned}$$

□

The approach of k-means is to find a partitioning of the data set in such a way that the sum of the variations is minimized.

Definition 3.4 (k-means)**.** *Let* $X = \{x_1, ..., x_n\}$ *be a data set and* $\{C_1, ..., C_k\}$ *a partition. Then k-means tries to find an optimal partition* $C^* = \{C_1^*, ..., C_k^*\}$ *such that:*

$$\sum_{l=1}^{k} D(C_l^*) = \min_{C_1, ..., C_k} \sum_{l=1}^{k} D(C_l) = \min_{C_1, ..., C_k} 2 \sum_{l=1}^{k} \sum_{i \in C_l} \sum_{j=1}^{p} (x_{ij} - \bar{x}_{lj})^2 \tag{3.2}$$

When it comes to solving the optimization problem defined by formula (3.2), the computational complexity of the algorithm to be used is of high interest. Therefore computational complexity theory categorizes problems into different classes that have some defining properties, one of which is called NP-hard. The definition of these classes would go beyond the scope of this work and therefore only a reference to the literature is given here e.g. [**np˙hard˙reference**]. In very simplified terms one can say that there are currently no efficient algorithms for this type of problem.

Corollary 3.2. *Solving the k-means problem defined in (3.2) is NP-hard.*

Proof. c.f. [**NP˙hard**] □

Even though the problem is NP-hard it is still possible to provide algorithms which converge to a local optimum. The algorithms used for finding a local minimum work on an iterative basis and involve just a few different steps. The k-means algorithms either start with an initial assignment of all observations to k different clusters or with k distinctly selected cluster centers. The next step is to find a new cluster center such that the variation $D(C_l)$ is minimized within each cluster. Then all data points $X = \{x_1, ..., x_n\}$ are reassigned to the cluster which is nearest and the minimization procedure is repeated until convergence.

Remark 3.4.

(i) For $k = n$, (3.2) is zero because each data point represents a cluster.

One possible formulation for an algorithm that converges to a local optimum is the one from Lloyd [**lloyd1982least**] which a pseudo code is given below. Note that for algorithm 1 we have a fixed number of clusters k as well as a finite set of possible partitions k^n. We can therefore show that the stated algorithm converges to a local minimum by minimizing (3.1).

Remark 3.5.

(i) For any set of observations S it holds that:

$$\bar{x}_S = \underset{x}{argmin} \sum_{i \in S} (x_i - x)^2$$

Hence, step 2a) minimizes the sum of squared deviations and therefore $D(C_l)$.

*(ii) Step 2*b*) which reassigns the observations to the new nearest centroid can only reduce the objective function.*

Algorithm 1 k-means clustering [**Introducion˙Stat˙Learning**] - Lloyd's algorithm

---

1. Choose k initial centroids randomly.
2. Iterate till the cluster assignments stop changing:
   a) For each of the k clusters, compute the cluster centroid. The i-th cluster centroid is the vector of the p parameter means for the observations in the ith cluster.
   b) Assign each observation to the cluster whose centroid is closest in the sense of Euclidean distance.

Remark 3.6.

(i) The k-means algorithm always converges and finds a local optimum which need not to be the global one.

(ii) Different clustering results can be obtained when different initial cluster assignments in step 1 of algorithm 1 are chosen.

Remark 3.7 (Running Time)**.**

(i) Algortihm 1 has a running time of $O(nkpi)$, with n being the number of data points, k the number of cluster, p the number of dimensions and i the number of iterations needed to converge. The only unknown variable is the number of iterations.

(ii) The trivial upper bound for the number of iterations needed is given by $O(k^n)$, because the algorithm visits every partition of points only once.

*(iii) It can be shown [***arthur2006slow***] that in the worst case scenario the running time of the algorithm is superpolynomial with a lower bound for the number of iterations of $O(2^{\Omega(\sqrt{n})})$. That means that the running time cannot be bounded above by any polynomial function.*

(iv) In practice, the number of iterations required is often small, which makes the algorithm appear to be linearly complex.

It is advisable to run the algorithm several times with different initial centroid assignments. This increases the likelihood of finding a partition that is

close to the optimum and thus provides a low value for the objective function (3.2). However, the biggest challenge when using k-means is the estimation of the optimal number of clusters k. Figure (3.1) shows an example with three clusters that illustrates the different clustering results when the number of k increases. Panel (a) shows the raw data with three clusters ($k_{true} = 3$), each generated by an uniform distribution. Panels (b), (c) and (d) show the cluster results with $k = 2, 3$ and 7, respectively. If the number of clusters k is smaller than the actual number of clusters in the data, then k-means merges clusters, while for k greater than k_{true}, k-means divides well seperated clusters. Panel (b) shows a merge of the clusters at the top right, whereas panel (d) shows an artificial split of two natural clusters for $k = 5$. For the grouping of similar data points and their representation by a cluster center, an underestimation of the actual number of clusters is more critical than an overestimation. If k underestimates the true value of clusters ($k < k_{true}$), then it's not possible to capture the cluster specific characteristics for the merged clusters because they are represented by only one cluster center. However, an overestimation of k_{true} is not so critical because some natural clusters will be represented by two cluster centers which has a negative impact on the compression ratio but not the clustering quality. For the most crucial step of k-means, a comprehensive collection of methods for estimating the correct number of clusters can be found in [**milligan1985examination**]. In the following, the 'elbow' and silhouette method, two widely used graphical methods for estimating k_{true}, are presented and applied to the sample data set.

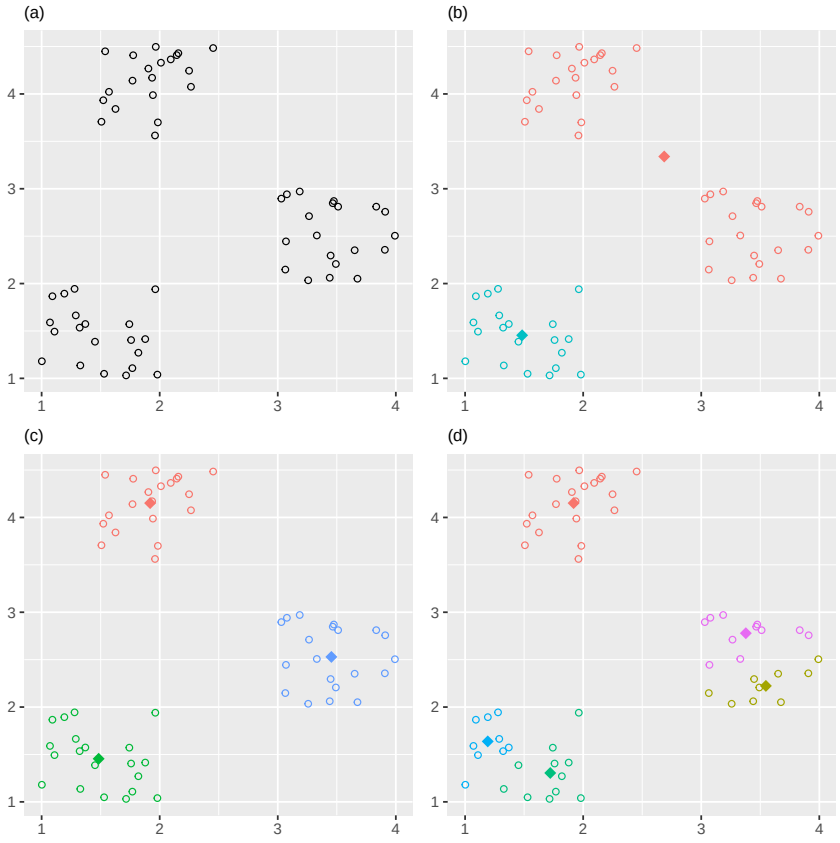


Figure 3.1.: Results for a three-cluster example: (a) raw data; (b) $k = 2$; (c) $k = 3$; (d) $k = 5$

3.1. Elbow method - gap statistic

Estimating k, the parameter that defines the number of clusters, is one of the most difficult tasks when using k-means. While it is still possible to graphically determine the number of clusters by plotting the data in 2 or 3 dimensional spaces, other methods must be used in higher dimensions. The 'elbow' method is one of the most commonly used methods in this context.

Definition 3.5. *Let $X = \{x_1, ..., x_n\}$ be a data set and k the number of clusters. Then $C = \{C_1, ..., C_k\}$ is the corresponding partition with the sum of variation within all clusters defined as:*

$$V_k := \sum_{r=1}^{k} \frac{D(C_r)}{2}$$

Plotting V_k, the sum of variation within all clusters as a measure of total error versus the number of clusters used gives a good indication for the true value of k. Figure 3.2(a) shows that the error measure V_k decreases monotonically as the number of clusters increases but there seems to exist a k from which the decline is clearly flattened. Such an 'elbow' indicates that any additional cluster reduces the total variation V_k only slightly and an appropriate number of clusters can be derived from the location of the 'elbow'. In accordance with $k_{true} = 3$, the 'elbow' plotted in 3.2(a) indicates that the true number of clusters is three, because there the curve starts to flatten dramatically. Even if the sample data set consists of very well separated clusters, the example indicates that the method could also be suitable in general use cases. For a small number of tasks the graphical determination of the "elbow" is practicable, but an automated method is needed with an increasing number of clustering operations. A statistical method that formalizes this procedure of finding the 'elbow' is described in [**tibshirani2001estimating**]. The basic idea is to make the sum of variation V_k comparable to a reference. For this purpose, the sum of deviations is calculated for each k and then compared with the expected sum of the deviations derived from a reference data set with no obvious clustering. The reference data set is generated by sampling uniformly over the range of the observed values for every feature from the original data set. This means it is sampled uniformly from the smallest p-dimensional cube that contains all data points $\{x_1, ..., x_n\}$ of the original data set.

Definition 3.6. *Let k be the number of clusters and $\mathbb{E}_n$ the expectation under a sample of size n from the reference distribution. Then the gap-statistic is*

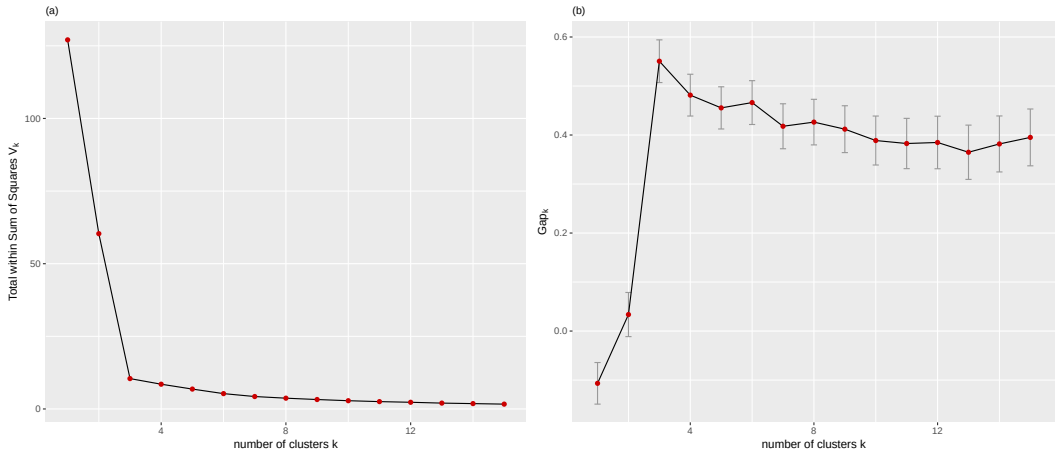


Figure 3.2.: Three-cluster example: (a) Total within Sum of Squares; (b) Gap-Statistic

defined as:

$$Gap_n(k) := \mathbb{E}_n\big[log(V_k)\big] - log(V_k) \tag{3.3}$$

To determine the expected value $\mathbb{E}_n\big[log(V_k)\big]$ we draw B different samples $\{x_1^*, ..., x_n^*\}$ from the p-dimensional cube and average over the B values of $log(V_k)$. Accounting for the simulation error introduced by using the B Monte Carlo samples the standard deviation is given by:

$$s_k = \sqrt{1 + \frac{1}{B}} sd(k)$$

The optimal cluster size k is then determined by the following rule which identifies the 'elbow':

Definition 3.7. *Let $Gap(k)$ be the statistic defined above and s_k the standard error (c.f. [**tibshirani2001estimating**]). Then the rule for choosing k is given by:*

$$\hat{k} := \textit{smallest } k \textit{ such that } Gap(k) \geq Gap(k+1) - s_{k+1} \tag{3.4}$$

In order to find the optimal number of clusters in a computational way, the following steps described in algorithm 2 are necessary:

Figure (3.2)(b) shows the gap-statistic for different values of k with the corresponding standard errors as a vertical bar. The application of the proposed rule

Algorithm 2 Ellbow method [**tibshirani2001estimating**]

1. Define the maximal number of clusters k_{max}
2. Iterate over $k = 1, ..., k_{max}$:
   a) Apply the k-means algorithm to the data set $\{x_1, ..., x_n\}$ and calculate $log(V_k)$.
   b) Generate B reference data sets, each of them having n samples $\{x_1^*, ..., x_n^*\}$.
   c) Cluster each of those B data sets and calculate $\mathbb{E}_n\big[log(V_k)\big] = \frac{1}{B}\sum_{b=1}^{B} log(V_{kb})$
   d) Compute the standard deviation:

$$sd(k) = \left(\frac{1}{B}\sum_{b=1}^{B}(log(V_{kb}) - \mathbb{E}_n\big[log(V_k)\big])^2\right)^{\frac{1}{2}}$$

   e) End the loop if: $Gap(k) \geq Gap(k+1) - s_{k+1}$

(3.4) for selecting the number of clusters results in $\hat{k} = 3$ which corresponds exactly to the actual number of clusters in the data set.

3.2. Silhouette method

Although the "elbow method" is well suited in many cases to determine the number of clusters, the resulting partitioning is not visually displayable if the dimension is larger than three. A visually appealing graphical display called silhouette plot introduced in [**rousseeuw1987silhouettes**] tries to overcome this shortcoming in order to be able to interpret cluster results more properly. The plot shows whether a specific partitioning result reflects a cluster structure actually present in the data set or not, by comparing the within dissimilarity with the between dissimilarity for every data point. It can thus be determined how similar a data point is to its own cluster compared to the other clusters. The measure of similarity can be calculated with any distance metric appropriate to the specific problem and will be called $dist(x_i, x_j)$ for two data points x_i and x_j.

Definition 3.8 (Within dissimilarity)**.** *Let $X = \{x_1, ..., x_n\}$ be a data set and $C = \{C_1, ..., C_k\}$ the corresponding partitioning with k cluster. Assume that*

the data point x_i is assigned to cluster C_i with $1 \leq i \leq k$. Then the average dissimilarity of x_i to all other objects assigned to cluster C_i is defined by:

$$WD(C_i, i) := \frac{1}{|C_i|} \sum_{x_l \in C_i} dist(x_i, x_l)$$

One can think of $WD(C_i, i)$ as a measure of how well the data point x_i is embedded in its cluster C_i. The smaller the value, the closer the data points of the cluster are to each other on average.

Definition 3.9 (Between dissimilarity)**.** *Let $X = \{x_1, ..., x_n\}$ be a data set and $C = \{C_1, ..., C_k\}$ the corresponding partitioning with k cluster. For any cluster C_j different from C_i (i.e. $i \neq j$) the average dissimilarity of $x_i \in C_i$ to the cluster C_j is given by:*

$$BD(C_j, i) := \frac{1}{|C_j|} \sum_{x_l \in C_j} dist(x_i, x_l)$$

One can see that $BD(C_j, i)$ is the average distance from data point x_i in cluster C_i to all data points x_j in cluster C_j.

Definition 3.10. *Let $X = \{x_1, ..., x_n\}$ be a data set and $C = \{C_1, ..., C_k\}$ the corresponding partitioning with k cluster. For any data point x_i assigned to cluster C_i (e.i. $x_i \in C_i$) the distance to the closest neighbor cluster is given by:*

$$dist(i) := \min_{C_i \neq C_j} BD(C_j, i)$$

Cluster C_b with $1 \leq b \leq k$, which shares the smallest average dissimilarity with point $x_i \in C_i$ is called the neighbor cluster of x_i. This neighbor cluster is the first choice if cluster C_i is removed from our analysis and we have to reassign x_i to a new cluster. After computing $WD(C_i, i)$ and $dist(i)$ for every data point $x_i,\ i = 1, ...n$ one can define the silhouette statistic $s(i)$ as follows.

Definition 3.11. *Let $X = \{x_1, ..., x_n\}$ be a data set and $C = \{C_1, ..., C_k\}$ the corresponding partitioning with k cluster. For every $x_i \in X$ the silhouette $s(i)$ is defined as:*

$$s(i) = \begin{cases} 1 - \frac{WD(C_i,i)}{dist(i)} & \text{if } WD(C_i, i) < dist(i) \\ 0 & \text{if } WD(C_i, i) = dist(i) \\ \frac{dist(i)}{WD(C_i,i)} - 1 & \text{if } WD(C_i, i) > dist(i) \end{cases}$$

Remark 3.8. *We can write the silhouette $s(i)$ in a more compact form:*

$$s(i) = \frac{dist(i) - WD(C_i, i)}{max\{WD(C_i, i), dist(i)\}}, \qquad \text{if } |C_i| > 1$$
$$s(i) = 0 \qquad \text{if } |C_i| = 1$$

Remark 3.9. *For every x_i it is true that $-1 \leq s(i) \leq 1$.*

Remark 3.10. *In order to understand which values of $s(i)$ correspond to which clustering results, it makes sense to look at values at the boundaries:*

- ⋆ *$s(i) \in (1-\epsilon, 1]$: For $s(i)$ close to 1 the within dissimilarity of x_i is much smaller than the minimum between dissimilarity. This shows on the one hand that the data point x_i is very well embedded in its cluster and has a small distance to the other data points of its cluster (i.e. $WD(C_i, i)$ small). On the other hand a relatively large value of $dist(i)$ shows that the minimum distance from x_i to the nearest cluster is very large. Thus one can speak of adequate clustering.*
- ⋆ *$s(i) \in (-\epsilon, +\epsilon)$: The within dissimilarity of x_i is approximately the same as the minimum between dissimilarity which indicates that x_i lies between two cluster. A small change of the data point x_i could cause it to be assigned to another cluster, suggesting unstable results.*
- ⋆ *$s(i) \in [-1, -1+\epsilon)$ The minimum between dissimilarity of x_i is much smaller than the within dissimilarity which indicates that x_i may not be correctly clustered. In this case x_i is on average much closer to the neighbor cluster than to the actual cluster which rises doubts if the cluster assignment is right.*

Remark 3.11. *Let x_i be a data point which is assigned to cluster C_i and cluster C_j the neighbor cluster of x_i. If x_i is reassigned from cluster C_i to cluster C_j then $s(i)$ becomes $-s(i)$*

In order to get a visually appealing overview if a clustering result is good or not all silhouettes are plotted on top of each other. Silhouettes which belong to the same cluster are plotted together and ranked in decreasing order.

Figure (3.3) shows the silhouette plot for the sample data set with $k = 3$ and 5, respectively. Each silhouette plots shows on the right side the cluster name, the number of data points assigned to the cluster and the average silhouette width. In the right panel of figure (3.3) cluster 1 consists of 11 data points

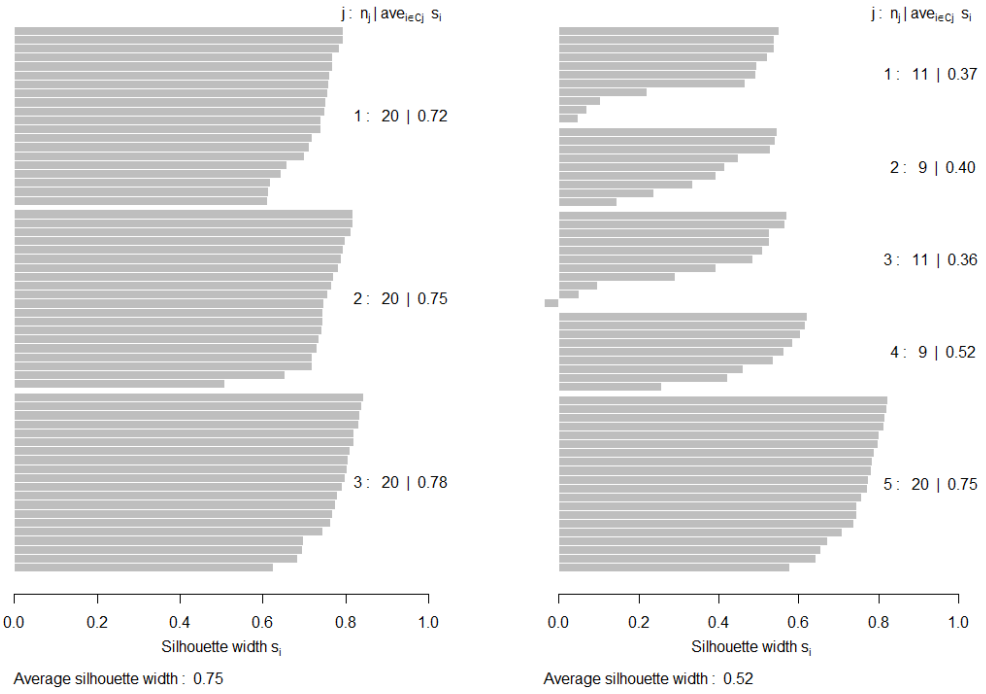


Figure 3.3.: Left: silhouette plot for $k = 3$; Right: silhouette plot for $k = 5$

and has a average silhouette width of 0.37. In cluster 3, for example, two data points have a silhouette width $s(i)$ close to zero and one data point has a negative value for $s(i)$. Except for cluster 5, all other clusters include data points which have a small or even negative value for $s(i)$. At the bottom of each silhouette plot the average silhouette width for all data points is given. The silhouette plot with $k = 3$ has much wider silhouettes compared to the plot with $k = 5$ which is an indicator that only 3 clusters are present in the data. Similar plots for $k = 2$ and $k = 4$ also lead to the conclusion that the data consists of 3 natural clusters. If there are too many ($k > k_{true}$) or too few clusters ($k < k_{true}$) some of the cluster silhouettes will be much narrower compared to the others.

Remark 3.12. *A high average cluster silhouette width indicates that the cluster is well separated from other clusters and is not split up artificially.*

Remark 3.13. *It can be seen that artificial splits of clusters gets quite heavily penalizes by the silhouette coefficient. The average silhouette width drops from 0.75 to 0.52 if the number of cluster is increased from $k = 3$ to $k = 5$ as seen in figure (3.3).*

3.3. Curse of dimensionality

After presenting two methods namely the elbow method and silhouette method which help identifying the correct number of clusters one should also be aware of phenomena that arise when analyzing data in high dimensional spaces. The 'curse of dimensionality' is a term introduced by Richard Bellman [**bellman1961adaptive**] to describe the rapid increase in volume and therefore the intractability of algorithms, when adding more dimensions of data to a mathematical space. Nowadays there are many different phenomenons referred to when talking about the curse of dimensionality but in the subsequent the focus is on distance functions as a measure of similarity. So far, all the methods presented are based extensively on the underlying distance functions used. To get a better understanding on the issue a simple example is given that helps to illustrate the problem.

Example 3.2.

(i) Imagine a line segment of length 1, and 10 data points which should represent the line. To capture the whole line segment one would distribute the points uniformly across the line. Therefore the line would be divided

into 10 segments with length $\frac{1}{10}$ and the points are centered within these segments.

(ii) By adding one dimension the line segment becomes a square segment with edge length 1. In order to represent the 'same' space with a data point as in the one-dimensional case, the square would have to be divided into 100 smaller squares with an edge length of $\frac{1}{10}$ each. In the center of each of those square segments one data point is needed as a representative. A total of 100 data points are required to represent the square segment as exactly as the line segment.

(iii) By adding another dimension the square segment becomes a cube and 1000 points are needed.

Example 3.2 illustrates that as the number of dimensions increases the number of data points rises exponentially in order to represent the whole space properly. Thinking of insurance data, one can have data points with various numbers of dimensions ranging from just a few to several hundreds dimensions. Considering the fact that the cash flow projections of grouped policies should coincide over a 60 year horizon, it is easy to see that only 5 of these cash flow characteristics (e.g. premium, costs, ...) result in data points with a dimension of 300. Of course, it is not advisable to include, over a period of 60 years, all cash flow variables in the grouping process, but it is much more difficult to find only those variables that are relevant for a major part of the results than including everything. For this reason, in practice more variables are often used than would actually be necessary. Another unintuitive fact one needs to be aware of is that the volume of a hypersphere inscribed into a hypercube is getting relatively smaller as the dimension increases.

Corollary 3.3. *Let a hypersphere with radius r and dimension d be inscribed into a hypercube with edges of length $2r$. Then we get for the proportion of the volumes:*

$$\frac{V_{Sphere}}{V_{Cube}} = \frac{r^d \frac{\pi^{\frac{d}{2}}}{\Gamma(\frac{d}{2}+1)}}{(2r)^d} = \frac{\pi^{\frac{d}{2}}}{2^d \Gamma(\frac{d}{2}+1)} \to 0 \text{ as } d \to \infty$$

Remark 3.14. *Corollary 3.3 says that as dimension d increases, more and more volume of the hypercube is outside the hypersphere. This means that under a uniform distribution most of the data points are located far from the center and thus close to an edge in a certain sense.*

Remark 3.15. *Some other examples why intuition fails in high dimensions are given in paragraph 6 of [***domingos2012few***].*

Not only do the data points move closer to the edge as the dimension d of the data space increases, but the distance between the individual data points is becoming more and more similar. It can be shown ([**beyer1999nearest**]) that under a broad set of conditions the distance to the nearest and to the farthest data point converges as dimensionality d increases. Experimental results in [**beyer1999nearest**] show that this effect can occur even for relatively low dimensional data with only 10 to 15 dimension. The fact that the distance to the farthest data point can get similar to the distance of the nearest data point makes clustering a hard job. The basic concept of k-means is to find, in the case of the Euclidean distance measure, spherically shaped clusters that have different characteristics. Therefore, data that is almost identical should not be clustered with a simple k-means algorithm without further analysis. Due to the fact that the k-means algorithm always returns a result even though there are obviously no clusters in the data because all data points are somehow similar, it is very difficult to determine whether k-means is a suitable tool for clustering or not. Situations in which all data points are similar and no cluster structure is present in the data are, as shown in the previous section, indicated by a low silhouette coefficient. When using the methods described in section 3.2, a silhouette plot with low silhouette coefficients for the clusters indicates that the clusters found by the algorithm are not well separated. Single clustering attempts can easily be verified by an visual inspection of plots described above. With an automated clustering approach, which is necessary for large insurance portfolios, visual control of the individual silhouette plots is not possible in most cases. In such cases, only a validation of the silhouette coefficient is feasible, but this leads ultimately to a situation where a clustering result is validated by a single value. It is therefore advisable to use low-dimensional policy data sets or conduct a thorough analysis of the data to avoid situations where problems referred to as 'course of dimensionality' occur.

4. Non-negative least squares (NNLS)

Appendix A.

Tables

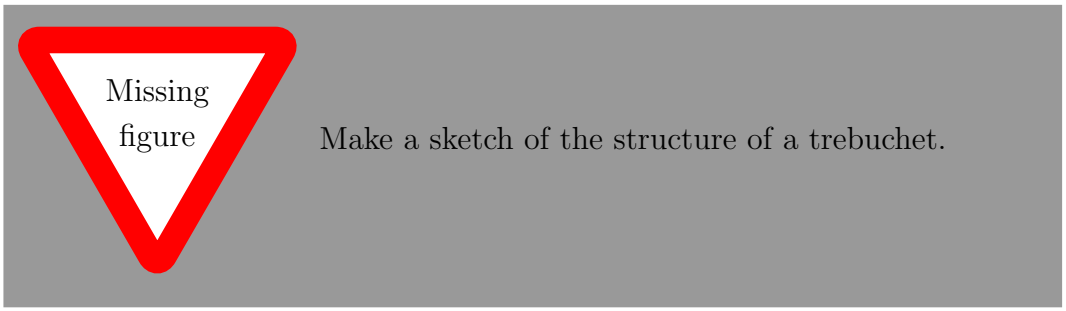
test

month	pop_15	pop_70	prem_15	prem_70	prem_diff	claims_15	claims_70	claims_diff
0	1	1						
9	0.999834	0.987259	385.60	1128.40	742.80	1.70	130.61	128.91
21	0.977140	0.947577	379.73	1092.63	712.90	4.73	185.70	180.97
33	0.952445	0.905554	370.17	1045.64	675.47	10.08	201.42	191.34
45	0.928273	0.863652	360.80	998.80	638.00	15.22	216.81	201.60
57	0.904661	0.821842	351.64	952.09	600.45	19.87	232.12	212.25
69	0.881638	0.780093	342.69	905.48	562.79	24.11	247.29	223.18
93	0.837332	0.696652	325.47	812.40	486.93	32.45	277.16	244.71
105	0.816021	0.654913	317.19	765.86	448.68	36.70	291.89	255.19
117	0.795253	0.613146	309.11	719.31	410.19	41.02	306.32	265.29
129	0.775013	0.571358	301.25	672.72	371.47	45.41	320.21	274.81
141	0.755288	0.529577	293.58	626.11	332.54	49.85	333.38	283.53
165	0.717332	0.446430	278.82	533.03	254.20	58.90	354.70	295.80
177	0.699076	0.405693	271.73	486.94	215.21	63.50	359.06	295.56
189	0.681284	0.365984	264.81	441.73	176.92	68.15	359.15	291.01
201	0.663944	0.327583	258.07	397.76	139.68	72.83	355.49	282.66
213	0.647046	0.290722	251.51	355.31	103.81	77.54	348.61	271.07
237	0.614523	0.222339	238.87	275.96	37.09	87.12	326.91	239.79
249	0.598856	0.191115	232.78	239.40	6.62	91.79	312.50	220.71
261	0.583562	0.162081	226.85	205.15	-21.70	96.52	295.62	199.10
273	0.568621	0.135426	221.05	173.38	-47.67	101.35	276.01	174.66
285	0.554015	0.111340	215.39	144.32	-71.07	106.29	253.63	147.35
297	0.539728	0.089980	209.85	118.16	-91.69	111.28	228.95	117.68
309	0.000000	0.000000	0.00	0.00	0.00	5983.32	991.18	-4992.14

Table A.1.: Yearly outputs for PVFP-sensitivity analysis based on an interst rate of 2%.

The todo text is the text that will be shown in the todonote and in the list of todos. The optional argument options, allows the user to customize the appearance of the inserted todonotes. For a description of all the options see sadf The todo text is the text that will be shown in the todonote and in the list of todos. The optional argument options, allows the user to customize the appearance of the inserted todonotes. For a description of all the options see The todo text is the text that will be shown in the todonote and in the list of todos. The optional argument options, allows the user to customize the appearance of the inserted todonotes. For a description of all the options see The todo text is the text that will be shown in the todonote and in the list of todos. The optional argument options, allows the user to customize the appearance of the inserted todonotes. For a description of all the options see fasd fadsf asdf

Make a cake . . .



Make a sketch of the structure of a trebuchet.