Erasmus University Rotterdam

Erasmus School of Economics

Visualization of Large-scale Shopping Cart Data using Correspondence Analysis

Master Thesis - M. Sc. Econometrics

Business Analytics and Quantitative Marketing

Author: Gerrit Alexander Krispien (484484)

Supervisor: Prof. Dr. P.J.F. Groenen

Second Assessor: Prof. Dr. M. van de Velden

Date Final Version: 19th December 2021

Acknowledgements

I would like to thank Professor Dr. P.J.F. Groenen (Erasmus University Rotterdam) for the great supervision and support when writing this master thesis, which did not suffer at all from a global pandemic as well as the fact that the time difference between our respective locations changed substantially multiple times.

Abstract

In this report the possibility of visualizing large-scale shopping cart data using Multiple Correspondence Analysis is investigated. The computational limits of Multiple Correspondence Analysis are found and a way to overcome those is suggested. A representative data set from an online grocery store is used which is agglomerated into three different data tables. The first one contains a limited amount of variable categories and allows for a traditional implementation of MCA once this is efficiently implemented. Next, a data table containing a large amount of variable categories is used and points out the computational limit of MCA for roughly 4,000 variable categories. An additional method, called PowerCA, is implemented which increases this limit drastically, but only gives an approximate solution. Data tables with a total amount of roughly 25,000 variable categories can be visualized using PowerCA within a reasonable amount of time. For even more variable categories, the idea of a meaningful grouping or subset of some of the variables is suggested in order to make the implementation of PowerCA feasible. This is also applied to the third data table given in this report, representing the given shopping-cart data set as a whole as it includes all relevant variables.

Contents

1 Introduction

Given the rise and ever-increasing volume of trade that is done through E-commerce in various product categories, webshops and companies selling goods online have the possibility to extract more and more data from their customers' online visits. This data is becoming more detailed than ever, thus describing the behavior of customers on online websites in great detail. For example, webshops can make profiles of their customers, track the items bought in general, generate fields of interest for each customer based on this and compare the purchasing behavior of different customers with similar interest. Further, patterns over time can be found such as seasonality patters which can even be made item-specific. More importantly, webshops can track which products are purchased together through co-purchases, enabling the creation of recommender systems in order to both offer the customer easier access to complementing products as well as increase sales. Lastly, good and detailed knowledge of the purchasing behavior of customers enables a webshop to calculate when it is best to restock and optimize inventory cycles.

At the heart of all of this lies the ability to extract this kind of information from the available data for which appropriate statistical methods are necessary. The choice of which methods to use directly impacts the quality of the results and hence the degree to which the above-mentioned advantages of these analyses realize themselves. It is hence of great importance to make this choice carefully.

One of the most direct sources of data for this stems from the online shopping cart of customers since it directly shows what kind of products were purchased together and when. Also, it is directly established from the general purchasing history of the online store. Given the huge number of products available in most online stores, the datasets which describe these shopping carts on a large scale are automatically of high dimension. It is hence not straight-forward to describe these datasets well in two- or three-dimensional space and extract valuable and representative information from them. Methods used for this problem generally have to be scalable to huge datasets in terms of computing resources and running time but most importantly, the results have to have a high degree of representativeness for the original patterns in the high-dimensional data.

Several research fields seem to offer approaches to this problem, such as the fields of dimension reduction (Van Der Maaten, Postma, and Van Den Herik 2009) as well as classical data visualization (Dietrich, Heller, and Yang 2015). Nonetheless, the application of these kinds of techniques to the specific case of online shopping cart data does not seem to be a well-researched problem, especially in a statistical context. Research fields such as Market Basket

Analysis seem to investigate the same problem, but only offer a rather broad Marketing or Management approach and do not go much into technical details (Blattberg, Kim, and Neslin 2008).

Furthermore, shopping cart data sets are often of categorical or qualitative nature, rather than quantitative. This limits the applicability of many dimension reduction techniques as it needs to be taken into account specifically. One of the most standard dimension reduction techniques for quantitative data is Principal Components Analysis (PCA) which does have a counterpart for the categorical case, namely Multiple Correspondence Analysis (MCA) (Greenacre and Blasius 2006). Hence, this seems like a suitable method but the specific application of MCA to the case of shopping cart data does not seem to have been investigated much. This is likely due to the fact that MCA uses the computationally expensive Singular Value Decomposition (SVD) and hence generally does not scale up well to large data sets. The main research question of this report can therefore be formulated as follows:

Does MCA represent a suitable method in order to visualize shopping cart data sets? If so, how far can it be scaled to large data sets and if not, what are ways to overcome its drawbacks and visualize large-scale shopping cart data?

This research paper will investigate several statistical methods, applied on a representative data set, in order to try to answer this question, starting with Multiple Correspondence Analysis (MCA). Next, a faster implementation of MCA, called PowerCA will be implemented as it seems to be able to handle larger amounts of data and circumvents the computationally expensive Singular Value Decomposition.

In the following sections a review of the corresponding literature is given, followed by a detailed description of the data set used in this analysis. Further, different subsets of the main data set are taken, each representing a specific use case for the given methods which are each described in the latter section as well. Next, the statistical theory of the implemented methods and necessary underlying concepts are discussed, followed by a detailed discussion of the computational results. Lastly, a summarizing conclusion is given.

2 Literature

The most famous technique used to reduce the dimensionality of a given dataset would be Principal Component Analysis (PCA) which was first developed in algebraic form as early as in 1933 by Harold Hotelling. He formalized the idea that a dataset could be described by a set of linear combinations while retaining as much of the variance as possible (Hotelling

1933). With the invention of computers, this algorithm could be applied to larger datasets and its applications in various fields grew rapidly. More methods with the same goal were developed, giving rise to the general field of dimensionality reduction (Jolliffe 2005). In more recent years, where computing power grew even more drastically and with the emergence of fields in engineering where extremely detailed measurements could be made, this field has been expanding quickly and several new methods have been introduced.

Generally, the field of dimensionality reduction can be classified into linear and non-linear dimensionality reduction. As the name suggests, linear dimensionality reduction techniques try to generate a best linear approximation in low-dimensional space of a high-dimensional dataset. The most famous method, PCA, being one of them. While PCA focuses on modeling the covariance structure of the data, other linear methods such as Factor Analysis (FA) focus on the correlation structure of the data (Ghodsi 2006). These methods are generally easily applicable, but suffer from the drawback of only giving a linear approximation. In contrast to this, non-linear dimensionality reduction techniques can overcome this drawback. Techniques developed in this field are among others Kernel PCA, Isomap and Local Linear Embedding. Due to their ability to handle complex, non-linear data techniques in this field have been shown to outperform their linear counterparts on artificial datasets. For a thourough overview, the reader is referred to Van Der Maaten, Postma, and Van Den Herik (2009).

Not many of these techniques have been investigated in the context of shopping cart data. Though the analysis of shopping behavior through shopping-cart data can be found in various other research domains, for example Iskandar, Shobirin, and Saputra (2017), Yang and Lai (2006) and Padhi, Mishra, and Kumar Dash (2012), the specific application of dimensionality reduction techniques to shopping cart datasets does not seem to represent a well-researched problem. As mentioned previously, a distinctive factor for the shopping cart data case is the categorical nature of the data, making many of the standard dimensionality reduction techniques inapplicable. The counterpart technique to PCA for categorical data would be Multiple Correspondence Analysis, which has an interesting history by itself.

Multiple Correspondence Analysis as it is known today is the result of a technique that has been invented multiple times by different authors independently. Initially in 1935, a technique to analyze the relations between two categorical variables has been formulated both by Hirschfeld (1935) and Horst (1935). They formalized this idea by representing the categories of the two variables in multidimensional space, which would today be known as Correspondence Analysis (CA). This idea has then been further developed to account for the relationships between more than two categorical variables, among others by Burt (1950). The parallel developments of these methods have led to various names for the same underlying

technique, such as optimal scaling, homogeneity analysis, dual scaling, biplot or MCA, just to name a few (Di Franco 2016).

Independently of this development, the same method was rediscovered in the French literature, by Benzécri (1969). This applies to both CA and MCA. Hence, one hence generally speaks of different schools of MCA. The French school of MCA has its roots in the aforementioned paper by Benzécri. A good example of a relatively recent summary of the main aspects of the Frensh school of MCA can be found in Le Roux and Rouanet (2010). The developments described in the previous paragraph are generally referred to as the Leiden school of MCA (Di Franco 2016). Examples of appropriate literature for the findings of this school are the works by Michael Greenacre, such as Greenacre and Blasius (2006) and Greenacre (2017), as well as Gifi (1990). This research paper mainly follows the literature of the Leiden school of MCA.

As already indicated in the introduction, a crucial step of the MCA algorithm is the computationally expensive Singular Value Decomposition which limits the applicability to large data sets. This problem has been addressed in the statistical literature only to some degree. Markos, Menexes, and Papadimitriou (2009) propose a scheme that seems to circumvent the implementation of the SVD to an inconveniently large matrix by directing the focus to the so-called Burt matrix[1], including some enhanced calculations. This does result in a procedure to find both row and column coordinates of the original data while only having to compute the SVD on a comparatively small matrix, but is only applicable to cases with a limited amount of total variable categories. Another recent approach is proposed by Iodice D'Enza, Groenen, and Van de Velden (2020), where the idea is to circumvent the SVD by using the Eigenvalue Decomposition instead. Then, an efficient statistical method, namely the so-called Power Method can be used to approximate the decomposition of the given matrix while drastically reducing the computational cost. The applicability of this method to the specific case of shopping-cart data has not yet been investigated and is an integral part of the present research paper.

3 Data

The data set used in this analysis stems from an American online grocery store called Instacart. The data set is comprised of 5 different relational tables which together contain the records of roughly 3.4 million orders made at the store by more than 200,000 users. For each order the position in the sequence of product orders by a given user is displayed, the time the order

[1]Note that the structure of the Burt matrix as well as other fundamental terms will be explained in detail in Section 4.

was made and the time in between the current and previous order. Further, a detailed record of all items that were placed in each order, the chronological order in which the shopping cart was filled as well as whether the item has been purchased before or not are given.

Table Name	Variable Name	Variable Type	Explanation
Orders	order_id	integer [1:3,421,083]	unique integer identifying each order
total entries: 3,421,083	user_id	integer [1:206,209]	unique integer identifying each user
	order_number	integer [1:100]	number of current order by specific user
	order_dow	integer [0:6]	indicating the weekday the order was made
	order_hour_of_day	integer [0:23]	indicating the time the order was made (rounded to hours)
	days_since_prior_order	integer [0:30] or -1	number of days between current and previous order by specific user -1 for the first order by that user
Order Products	order_id	integer [1:3,421,083]	see above
total entries: 33,819,106	product_id	integer [1:49,688]	unique integer identifying each product
	add_to_cart_order	integer [1:145]	position the products was put into the basket
	reordered	boolean	whether the product was bought before or not
Products	product_id	integer [1:49,688]	see above
total entries: 49,688	product_name	string	name of the product
	aisle_id	integer [1:134]	unique integer identifying each aisle
	department_id	integer [1:21]	unique integer identifying each department
Aisles	aisle_id	integer [1:134]	see above
total entries: 134	aisle	string	name of the aisle
Departments	department_id	integer [1:21]	see above
total entries: 21	department	string	name of the department

Table 1: Instacart Shopping Cart Data Set

Given the fact that the data stems from a grocery store, the total number of different items is roughly 49,000 which are organized in 124 different aisles of different categories. These are themselves placed into 21 different departments (Instacart 2017). From this point onward, the Instacart Shopping Cart Data Set will be referred to as ISCD. A detailed overview of the different tables together with their variables, as well as the number of categories per variable is displayed in Table 1.

The data set can be agglomerated into one large table containing roughly 33.5 million observations of 11 variables. The need for proper statistical methods reducing this extremely large data set into the factors that most describe the relevant information contained in it becomes clear.

In this analysis, different methods are implemented with the overall goal of visalizing the information contained in the ISCD. Given the sheer size and complexity of the data present, it is obvious that different approaches for the same methods are possible, depending on how the different tables are agglomerated and to which variables the focus is directed to. It is the goal of this analysis to explore these different approaches appropriately and give an overview of how the given methods can be used on the ISCD. Hence, subsequently the reader is introduced to different combinations and subsets of the ISCD which are used in this report.

3.1 Data Table for Traditional MCA

As a starting point, the ISCD is put into its most basic form where it is reduced to the most central variables. This results in a table containing variables with a limited amount of total categories. More specifically, a subset from the **Orders** table is taken where the unique `order_id` is removed, given that when focusing only on this table it is a mere index over all entries in the table. Further, the `user_id` is removed as well since otherwise this variable introduces more than 200,000 new categories which for a first traditional implementation of MCA seems inappropriate. Note that in this way the information of which orders belong to the same user is partly lost, but that the variable `order_number` still contains part of this information as it clearly shows some orders which were not made by the same user[2].

Hence, this first table for the traditional implementation of MCA contains all orders made by all users, described by the following variables: `order_number`, `order_dow`, `order_hour_of_day` and `days_since_prior_order`. The total number of observations is equal to roughly 3.4 million while the total number of variable categories equals 163. Here the intention is to

[2]As an additional explanation on this, all first orders were made by different users, all second orders were made by different users, all third orders as well and so on.

find relationships and information in the data such as whether the order number relates to a specific time or weekday, or whether repeated orders are rather made by new customers or experienced customers, just to name a few.

3.2 Data Table containing many categories

Next, an agglomeration of **Orders** and **Order_Products** is made. This can be done in a relatively straight forward way by noting that the **Order Products** table simply contains additional observations for each order from the **Orders** table, linked by the variable `order_id`. Therefore, the variables of the two tables are joined together in one larger table, where all values from **Orders** are repeated for the corresponding observations in **Order Products**. This is equivalent to joining the two tables with a so called *inner join* in the SQL database language on the common variable `order_id`.

Again, in order to increase the total number of total variable categories gradually the same variables as described in the previous subsection are removed from the **Orders** table. Hence the resulting table contains the same four variables as well as `product_id`, `add_to_cart_order` and `reordered`. The total number of observations is equal to roughly 33.8 million while the number of total variable categories is equal to almost 50,000. Here the intention is to find similar relationships and information as in the first simple table, but now in more detail containing information about individual products in the order. Hence, whether certain products are bought at specific times or weekdays, whether some products correspond to new or experienced customers or whether reordered products relate to frequent or infrequent orders are relationships that are expected to be found in this data table.

3.3 ISCD Data Table

Lastly, a data table is created that represents the entire ISCD as a whole. Given that the previous data tables were subsets of the ISCD, it remains to attempt to aggregate all tables from the ISCD in one data table. In order to achieve this one should first note that the last three data tables listed in Table 1 should not be included simultaneously. This is due to the fact that `product_name` is a mere description of `product_id` and that `aisle_id` and `department_id` represent groupings of the actual products. They hence give more information when interpreting a solution but should not be included in one data table as separate variables. It is chosen to include information of the products through their unique `product_id`.

This results in the need to aggregate the two tables **Orders** and **Order Products** which

is done in the same way as in Section 3.2, namely through the common variable `order_id`. Additionally, the variables `user_id` and `order_id` are included as well. Note that both of these variables have an immense amount of variable categories (roughly 206,000 and 3,400,000, respectively) which gives the need for summarizing them in a way that results in a lower number of total variable categories. Therefore, the users are grouped in terms of the number of total orders each user made, and the orders are grouped in terms of the size of each order, where the size refers to the number of products in the basket. This reduces the number of variable categories for the variables `user_id` and `order_id` to 97 and 113, respectively. This results in a total of 9 categorical variables which consist of all variables present in both the **Orders** and **Order Products** tables, where `order_id` is only included once. From this point onward, the two grouped variables will be referred to as `users` and `orders`. This data table hence has around 50,000 total variable categories with roughly 33.8 million observations.

Note that the choice of grouping the two additional variables in this way leads to a certain relation between `users` and `order_number` as well as `orders` and `add_to_cart_order`. Namely, `order_number` indicates the position of a given order within all orders made by a certain user and `users` gives the total number of orders made by that user. The values of `users` hence equal the maximum value of `order_number` for each user. The same relation applies to the other pair of variables in terms of products. Nonetheless, the intention is to reveal separate information given that `users` distinguishes frequent from unfrequent customers while `order_number` contains information about the position of each order within the sequence they were made. The same reasoning applies to the other pair of variables for the total size of an order compared to the sequence the products were put into the basket.

4 Methodology

Two methods are applied in this analysis, both with the aim of visualizing the high-dimensional data set in a lower dimensional space as representative as possible. The methods vary in their computational complexity and in the amount of data they can process in a reasonable amount of time. First, Multiple Correspondence Analysis (MCA) is applied as it represents the counterpart to PCA for categorical data (Greenacre and Blasius 2006). Next, in order to account for the large amount of data present in the given shopping-cart dataset, a fast version of MCA, called PowerCA is implemented (Iodice D'Enza, Groenen, and Van de Velden 2020). In this section the theory of the methods as well as related concepts are explored in detail.

4.1 Singular Value Decomposition

Central to the methods used in this report is the Singular Value Decomposition (SVD), which represents a classic technique that would now be classified in the field of Linear Algebra. Early contributions to this technique date back to the 19^{th} century, while the invention of efficient algorithms for its computation dates back to the mid 1900s. It is a fundamental matrix factorization technique which is used in many algorithms in dimension reduction, such as Principal Component Analysis (see Stewart (1993) for a historical perspective). The SVD Theorem states that any numerical matrix $A \in \mathbb{R}^{m \times n}$ can be factorized into three matrices $U \in \mathbb{R}^{m \times m}$, $V \in \mathbb{R}^{n \times n}$ and Σ such that

$$A = U \Sigma V^T, \tag{1}$$

where both U and V are square, orthogonal matrices and $\Sigma = diag(\sigma_1, ..., \sigma_p)$ with $\sigma_1 \geq \sigma_2 \geq ... \geq \sigma_p \geq 0$. This implies that we have $U^T U = I_m$ and $V^T V = I_n$. Note that this definition only holds for real matrices A, but that generally the Singular Value Decomposition also holds for the complex case. In Equation 1 the elements of Σ are called singular values and the rank of A equals the number of non-zero singular values. For a full display of the complex case as well as a proof of the SVD Theorem the reader is referred to Klema and Laub (1980).

Many algorithms and software packages exist for implementing the SVD, the standard one for the R programming language being the package `svd`. Generally the computation of the SVD requires as lot of computational power for large matrices A as it has a computational complexity of $O(m^2 n)$ (assuming $m \geq n$) (Golub and Van Loan 1996). Modern, efficient approaches, such as the Lanczos bilinear diagonalization method (which is implemented in the R package `svd`) exist, but nonetheless, the SVD represents a bottleneck in many algorithms where it is applied when dealing with matrices of substantial size (Korobeynikov and Larsen 2019).

4.2 Multiple Correspondence Analysis

Multiple Correspondence Analysis is a classical dimensionality reduction technique that allows to analyze the relationship of multiple categorical variables. It represents the extension from regular Correspondence Analysis (CA), which analyzes the relation between two (sets of) categorical variables only, to the relations between multiple categorical variables. This is achieved by applying the regular CA algorithm to the so-called Superindicator matrix G or

the Burt matrix B which are established from a data table with multiple categorical variables as columns and observations as rows. Let G_j be the indicator matrix with dummy coding for variable j. Then G equals the concatenation of these indicator matrices for each variable, namely $G = [G_1, G_2, ..., G_j, ..., G_m]$ for $1 \leq j \leq m$. Thus, G becomes a large, sparse, binary matrix and B becomes a symmetric matrix of all cross tabulations of the categorical variables.

The result can be displayed in a lower dimensional space along so called principal axes, in which both all variables and all observations can be represented as a point. The overall pattern of these points can then be analyzed and used to draw conclusions about possible relationships between the variables as well as groups of observations. The principal axes, representing newly defined dimensions, are calculated in such a way that, in decreasing order, the axes display as much variation from the original data set as possible. This is also called the inertia of each dimension. This gives an indication of how much variation from the original data set is displayed in the lower-dimensional plot and hence of its quality (Greenacre and Blasius 2006).

MCA can be expressed in various different ways and has been reinvented multiple times in the literature (Greenacre 1988). Relevant to this analysis are the derivation which expresses the solution as the minimization of a loss function as well as the resulting algorithm to compute the MCA solution. Both parts are explained in the following subsections.

4.2.1 CA Loss Function

In order to outline CA and hence MCA defined by minimizing a loss function it is useful to revisit the basic Eckhart-Young theorem. Let Y be a data matrix of rank S. The Eckhart-Young theorem states that the residual sum of squares, when approximating Y by a rank-s matrix $\hat{Y}$, given by $\|Y - \hat{Y}\|^2$, is minimized by setting $\hat{Y}$ equal to the SVD of Y, namely $\hat{Y} = U\Sigma_s V^T$ (Eckart and Young 1936). Note that in this notation $U\Sigma_s V^T$ represents the standard SVD of Y, but with $\sigma_i = 0 \quad \forall\, i > s$, where σ_i represents the i^{th} diagonal element of Σ. Hence, $\hat{Y}$ results in a rank-s approximation of Y.

This result plays a central role in CA, as it is used to minimize the loss function, which can be expressed in several ways. Note that in this report the perspective of a matrix of probabilities, in accordance with Greenacre (2017), is followed. Other perspectives such as the approximation to the deviations from the independence model are possible as well, as outlined in Gower, Groenen, and Van De Velden (2010). Let W be the input matrix of interest containing only non-negative elements with dimensions $[r \times c]$ and let n be its grand total, meaning $n = \mathbf{1}^T W \mathbf{1}$. Note that $\mathbf{1}$ indicates a vector of ones of appropriate length. Further,

let P be the *correspondence matrix* defined by $P = n^{-1}W$ and let $\vec{r}$ and $\vec{c}$ be the row and column totals of P, respectively. That is, $\vec{r} = P\mathbf{1}$ and $\vec{c} = P^T\mathbf{1}$. Finally, let D_r and D_c denote diagonal matrices with $\vec{r}$ and $\vec{c}$ on their diagonal, respectively. MCA is concerned with approximating the deviations from W, the residuals, but it turns out that the standardized residuals are of more interest. Hence one computes the so called *standardized residuals matrix* $S = D_r^{-\frac{1}{2}}(P - \vec{r}\vec{c}^T)D_c^{-\frac{1}{2}}$ (Greenacre 2017). Thus, one is concerned with minimizing the loss function

$$\|D_r^{-\frac{1}{2}}(P - \vec{r}\vec{c}^T)D_c^{-\frac{1}{2}} - \hat{W}\|^2. \tag{2}$$

As explained in the previous paragraph, this is achieved by computing the SVD of S

$$S = D_r^{-\frac{1}{2}}(P - \vec{r}\vec{c}^T)D_c^{-\frac{1}{2}} = U\Sigma V^T. \tag{3}$$

One could directly base the final solution on this result by plotting the first s columns of $U\Sigma^{1/2}$ and $V\Sigma^{1/2}$, for a s-dimensional approximation. Though, it seems like it is much more common to rewrite the loss function as

$$\|D_r^{\frac{1}{2}}[D_r^{-1}(P - \vec{r}\vec{c}^T)D_c^{-1} - \hat{W}]D_c^{\frac{1}{2}}\|^2, \tag{4}$$

which represents a least squares problem with weights $D_r^{\frac{1}{2}}$ and $D_c^{\frac{1}{2}}$. Given that in Equation 4, $\hat{W}$ now approximates $D_r^{-1}(P - \vec{r}\vec{c}^T)D_c^{-1} = D_r^{-\frac{1}{2}}U\Sigma V^T D_c^{-\frac{1}{2}}$, using the solution from Equation 3 one generally bases s-dimensional plots on the first s columns of $F = D_r^{-\frac{1}{2}}U\Sigma$ and $H = D_c^{-\frac{1}{2}}V\Sigma$. Note that F represents the rows and H the columns of the original data in W (Gower, Groenen, and Van De Velden 2010).

4.2.2 MCA Computation

The previous results lead to a neat algorithm to compute the MCA solution from a data matrix W which is outlined subsequently. Note that depending on whether W is established from a data table containing two or multiple categorical variables, the method would be called CA or MCA, respectively. For this report the case of multiple categorical variables is relevant and hence the name MCA is used. As indicated earlier, the MCA solution can be computed in two ways which give the same plot in lower dimensions (though scaled) but slightly different inertias. MCA on the Superindicator Matrix G is computed by applying the regular CA algorithm to G and MCA on the Burt Matrix $B = G^T G$ is computed by applying

the regular CA algorithm to B. The principal inertias of the indicator version are the square roots of those of the Burt version. The reader is referred to Greenacre and Blasius (2006) for details.

Note that due to the shape of B, traditional MCA can already handle vast amounts of data for the case of an extremely large number of observations, but a limited amount of variable categories. Once the (possibly computationally expensive) part of creating B has been implemented, the CA algorithm applied to B itself only requires a limited amount of computational power and time, which remains fixed regardless of the amount of observations. This will be the starting point of the application of MCA in this analysis.

The derivation outlined in the previous subsection results in the following algorithm for computing the MCA solution from an input matrix W. Note that depending on the choice of type of MCA, W will be set equal to either G or B.

Algorithm 1: CA Algorithm

Input : Input matrix W

Output: MCA solution

1 calculate $n = \mathbf{1}^T W \mathbf{1}$, $P = n^{-1}W$, $\vec{r} = P\mathbf{1}$ and $\vec{c} = P^T\mathbf{1}$

2 establish the diagonal matrices D_r and D_c by using $\vec{r}$ and $\vec{c}$, respectively

3 calculate the *standardized residuals matrix* $S = D_r^{-\frac{1}{2}}(P - \vec{r}\vec{c}^T)D_c^{-\frac{1}{2}}$

4 compute the SVD, $S = U\Sigma V^T$

5 compute the *principal coordinates* of the rows and columns, $F = D_r^{-\frac{1}{2}}U\Sigma$ and $H = D_c^{-\frac{1}{2}}V\Sigma$, respectively

6 return F, H

Note that B is a symmetric matrix and hence when applying the above algorithm to $W = B$, the coordinates in F and H are equal. For a detailed description and explanation of the CA algorithm the reader is referred to Greenacre (2017), Appendix A.

4.2.3 Principal Inertia

As already indicated at the beginning of section 4.2, the principal axes in the CA solution are ordered by the amount of variation they capture from the original data set, in decreasing order. This means that the most influential dimensions come first and the higher the dimensions being analyzed become, the less representative they become. This is quantified by the so-called *inertia* of each dimension, denoted by $\phi_k = \sigma_k^2$ where σ_k represents the kth diagonal element (singular value) of Σ, calculated in line 4 of Algorithm 1. Additionally, the relative inertia

$\frac{\phi_k}{\sum_k \phi_k}$ gives a measure of how representative dimension k is within the overall CA solution (Greenacre and Blasius 2006).

The principal inertias of the dimensions are themselves further decomposed over the rows and columns of the original input matrix W. Namely, the principal inertia ϕ_k is decomposed into contributions of each row i as $\phi_k = \sum_i r_i f_{ik}^2$. Note that r_i represents the ith element of the vector $\vec{r}$ introduced in section 4.2.1 and f_{ik} represents the principal coordinate of point i on dimension k, hence an element of the matrix F, calculated in line 5 of Algorithm 1. In the same way ϕ_k can be decomposed along the columns of W as $\phi_k = \sum_j c_j h_{jk}^2$ where c_j is an element of $\vec{c}$ and h_{jk} an element of the matrix H.[3] These lead to useful quantities that can be used to further interpret and analyze a given CA solution.

The first ratio $R_1^{(ik)}$ takes the contribution of row i relative to the principal inertia of dimension k, namely

$$R_1^{(ik)} = \frac{r_i f_{ik}^2}{\phi_k}. \tag{5}$$

It can hence be used to find the most important points of dimension k. This makes the interpretation of the results easier as one can focus on certain points, which is especially useful for the case of many observations or many variable categories. Additionally, this information allows for a general interpretation of the given dimension.

The second ratio $R_2^{(ik)}$ takes the contribution of row i to dimension k relative to the total inertia of row i, namely

$$R_2^{(ik)} = \frac{r_i f_{ik}^2}{\sum_k r_i f_{ik}^2} = \frac{f_{ik}^2}{\sum_k f_{ik}^2}. \tag{6}$$

It hence indicates how well this point is represented in the given dimension k and allows for determining whether a given point can be interpreted with confidence or not. This measure can further be interpreted geometrically as so-called squared cosine or squared correlation.[4]

The indicated measures can be computed in a straight-forward way from a CA solution and offer more flexibility in its interpretation. This becomes especially useful when dealing with large data sets and many rows and columns since they help in determining the most important points. For details on the geometric interpretation as well as a derivation of the results the reader is referred to Greenacre (2017).

[3] Note that in the case where W equals the Burt matrix B, the components of the row and column decompositions of ϕ_k are equal.

[4] Note that both ratios can be computed for the column contributions as well by replacing the corresponding terms from the row to column decompositions, as mentioned previously.

4.3 PowerCA

The PowerCA algorithm, developed by Iodice D'Enza, Groenen, and Van de Velden (2020), represents a fast version of CA, given that the Singular Value Decomposition (SVD) in the regular CA algorithm becomes intractable quickly as the size of the input matrix W grows. This method is based on the so called Power method, a relatively old method for finding the most dominant eigenvector and its corresponding eigenvalue of a diagonalizable matrix.

4.3.1 The Power Method

The Power Method represents an old, well-known technique for finding a solution to eigenvalue problems. It is based on the fact that when multiplying a matrix $C \in \mathbb{R}^{k \times k}$ with any k-dimensional, non-zero vector α, the contribution corresponding to the largest eigenvalue of C increases more than the contribution of smaller eigenvalues (in absolute sense). This leads to the fact that when repeating this multiplication a large number of times, one gets a sequence of vectors $\alpha^{(h)}$ which converges to the largest eigenvector of the matrix C. Note that the vectors in $\alpha^{(h)}$ have to be normalized appropriately. This procedure can hence be used to extract the largest eigenvector eigenvalue pair from the matrix C. For pseudocode of this algorithm as well as proof of convergence, the reader is referred to Saad (2011), Chapter 4.

4.3.2 Application to CA

Before explaining the idea of using the Power Method within the computation of CA, it should be explained how it can be used to find not just the most dominant eigenvector eigenvalue pair, but the s most dominant eigenvector eigenvalue pairs. Note that the notation used is such that the sequence $(\alpha_1, \lambda_1), (\alpha_2, \lambda_2), ..., (\alpha_s, \lambda_s)$ is ordered from most dominant to least dominant and that this implies $\lambda_1 \geq \lambda_2 \geq ... \geq \lambda_s$.[5] It can be done in a relatively straight-forward manner by noting that when one subtracts from the matrix C its rank-$(s-1)$ approximation, one can apply the Power Method to that matrix in order to find (α_s, λ_s). The rank-$(s-1)$ approximation of C is in turn given by $\sum_{i=1}^{s-1} \lambda_i \alpha_i \alpha_i^T$, resulting in the following procedure for obtaining $(\alpha_1, \lambda_1), (\alpha_2, \lambda_2), ..., (\alpha_s, \lambda_s)$ from the matrix C (Husson et al. 2019):

[5]Logically, in this notation α_s refers to the eigenvector and λ_s to the corresponding eigenvalue.

Algorithm 2: Obtain the s most dominant eigenvector eigenvalue pairs from C

Input : Symmetric matrix C

Output: $(\alpha_1, \lambda_1), (\alpha_2, \lambda_2), ..., (\alpha_s, \lambda_s)$

for *i in* $1, ..., s$ **do**
    apply the `Power Method` on C to calculate (α_i, λ_i)
    $C = C - \lambda_i \alpha_i \alpha_i^T$
end
return $(\alpha_1, \lambda_1), (\alpha_2, \lambda_2), ..., (\alpha_s, \lambda_s)$

This computationally efficient procedure can be incorporated in the CA computation in the following way. Note that the SVD of S in Equation 3 can also be obtained via two Eigenvalue Decompositions (EVD) of SS^T and S^TS. Using the same notation this implies

$$SS^T = U\Lambda U^T \qquad \text{and} \qquad S^TS = V\Lambda V^T, \tag{7}$$

where U and V represent the same matrices as indicated in Section 4.1 and $\Lambda = \Sigma^2$. Note that Λ by itself is the diagonal matrix of eigenvalues. Hence one can proceed to compute the low-dimensional representation of the rows and columns of the original input matrix X by computing $F = D_r^{-\frac{1}{2}} U \Lambda^{\frac{1}{2}}$ and $H = D_c^{-\frac{1}{2}} V \Lambda^{\frac{1}{2}}$, respectively (Iodice D'Enza, Groenen, and Van de Velden 2020).

Having this relation of CA to the EVD in mind, one can construct a fast version of CA by replacing the EVD with the procedure outlined in Algorithm 2. This leads to the following schematic description of the overall algorithm with the name PowerCA, applied to a given input matrix W:

Algorithm 3: PowerCA Algorithm

Input : Input matrix W

Output: PowerCA solution

calculate $n = \mathbf{1}^T W \mathbf{1}$, $P = n^{-1}W$, $\vec{r} = P\mathbf{1}$ and $\vec{c} = P^T\mathbf{1}$
establish the diagonal matrices D_r and D_c by using $\vec{r}$ and $\vec{c}$, respectively
calculate the *standardized residuals matrix* $S = D_r^{-\frac{1}{2}}(P - \vec{r}\vec{c}^T)D_c^{-\frac{1}{2}}$
use Algorithm 2 on the matrix S in order to calculate $(\alpha_1, \lambda_1), (\alpha_2, \lambda_2), ..., (\alpha_s, \lambda_s)$
compute the *principal coordinates* of the first s dimensions of the rows and columns, $F = D_r^{-\frac{1}{2}} U \Lambda^{\frac{1}{2}}$ and $H = D_c^{-\frac{1}{2}} V \Lambda^{\frac{1}{2}}$, respectively
return F, H

Note that here one also has the choice between the two versions of MCA, again by chosing to set W equal to either G or B. In case $W = B$, the coordinates in F and G are again equal. The above algorithm runs much faster and has a lower computational complexity than the regular MCA algorithm shown in Algorithm 1 since the computationally expensive SVD is circumvented. Note that the solution is only an approximation as the Power Method only approximates the most dominant eigenvector of a given matrix. Nonetheless, it represents a strong solution for computing a lower-dimensional representation of a large, complex data set and will be heavily used in this report.

5 Results

This section presents the computational results of the described methods on relevant data tables of the ISCD. It will be shown when the methods themselves reach their computational limit and how they compare between each other on data tables of different sizes. Further, the actual computational results on the given data tables will be analyzed and interpreted as well. The applied methods consist of the regular CA algorithm which will be implemented both in a standard, pre-programmed R package, namely FactoMineR, as well as a manual implementation. The manual implementation corresponds to Algorithm 1 described in Section 4.2. Further, the PowerCA algorithm as described in Section 4.3 will be applied as well. It will be mentioned which specific data table from Section 3 is concerned in every subsection. The methods were all run on the same computer with an AMD Ryzen 7 4700U 8-core processor, 16GB random access memory and an SSD hard drive.

5.1 Comparison on Traditional Data Table

As a starting point, the methods are implemented on a data set in rather traditional form, with a limited amount of total variable categories. This allows for a rather standard implementation of the methods via the Burt matrix B. The size of the Burt matrix then remains constant, despite the number of observations in the data table. This is useful in order to test the three methods on different numbers of observations, which are gradually increased. It allows for testing whether their results match as well as comparing them with respect to their running times. For this first implementation of the methods, the core variables from the **Orders** table were taken, as described in Section 3.1. Note that the categories of these variables are described in Table 1, Section 3 and that appropriate histograms display the frequencies of the

variable categories in Appendix A, Figures 9 - 12. The running times for different numbers of observations are compared in Table 2. Note that for all numbers of observations listed in this table, the size of the Burt matrix is equal to 163 $\times$ 163.

observations	**FactoMineR**	**Manual MCA**	**PowerCA**	**calculate B**
100,000	27.61	0.56	0.49	1.42
1,000,000	438.28	0.59	0.52	18.87
3,421,083	-	0.59	0.55	76.25

Table 2: Running times of the methods on traditional data, in seconds

General comparisons between the methods can be derived from Table 2. As expected, FactoMineR takes the longest for all amounts of observations. This is likely due to the fact that not only the core MCA result is calculated, but also several other additional results such as value tests for the variable categories and squared cosine values, just to name a few. Further, all results are programmed neatly into one object from which all sorts of plots can be made, such as a biplot, graph of individuals or a graph of variables and/or their categories.[6] Though very complete and accurate, this limits the applicability to large data sets. As seen in Table 2, with more than 1 million observations on this comparatively easy data table FactoMineR reaches its limits and was not able to give a solution within a reasonable amount of time.

In contrast to this, the manual implementation of MCA as well as PowerCA were able to compute solutions within seconds. Note that the time to calculate the Burt matrix B is displayed separately since this step took the longest. This is clearly due to the fact that on this data table the number of total variable categories is limited and hence once the Burt matrix is calculated from the data, the two methods run in a known, predictable way. More specifically, on this data table the total number of categories equals 163, resulting in a 163 times 163 matrix B, which is nothing out of the ordinary. Logically, with more observations the time to calculate B increases, but the actual methods run similarly fast, despite the number of observations.

5.1.1 Computational Comparison

In order to give a first comparison between all three implemented methods, the results of the second row in Table 2 are compared in more detail. First, it is of interest whether the solutions are comparable. More specifically, the solutions of FactoMineR and the manual

[6]See http://factominer.free.fr for more details.

implementation of MCA should be in fact equal (up to a certain threshold) given that the same underlying MCA algorithm is implemented. Further, their solutions compared to the solution of PowerCA should match as closely as possible, given that PowerCA approximates the original MCA solution. Table 3 shows the Root Mean Square Error (RMSE) of the variable coordinates of both the manual MCA implementation and PowerCA versus the solution of FactoMineR. The RMSEs are taken per dimension of the solutions, for the first five dimensions.

RMSE against FactoMineR	**Dim 1**	**Dim 2**	**Dim 3**	**Dim 4**	**Dim 5**
Manual MCA	8.34e-12	2.90e-11	7.80e-11	2.36e-10	6.14e-10
PowerCA	2.11e-06	3.48e-06	1.06e-05	1.98e-01	4.84e-01

Table 3: RMSE of solutions, per dimension

One can infer that the manual MCA implementation is correctly implemented as the variable coordinates are identical to the solution of FactoMineR for all five dimensions up to an RMSE of less than 10^{-9}. This gives confirmation that this implementation can safely be used on larger data sets and taken as a reference for the original MCA solution as FactoMineR breaks down much earlier. Next, the solution of PowerCA shows a smaller degree of similarity which is still more than acceptable for the first three dimensions, given that the RMSE values are well below 10^{-4}. For dimensions 4 and 5 the RMSEs increase substantially. This can be explained by the fact that PowerCA only approximates the original MCA solution given that the Power Method only approximates the most dominant eigenvalue eigenvector pair of the corresponding standardized residuals matrix S. Given that the second most dominant eigenvalue eigenvector pair is based on the first one, the approximation error accumulates with higher dimensions. One should hence keep in mind that PowerCA never gives an exact solution and also that the higher the dimensions are which are being analyzed, the less accurate the inferences of the solution become.

From this point onward, for larger data tables of the ISCD, it will be assumed that the solutions of the three methods are comparably similar and not all three solutions will be displayed or compared in terms of their RMSE. The most accurate solution that is available will be displayed and analyzed. Given that FactoMineR already broke down rather quickly on this traditional subset of the ISCD, whenever the solution of the manual MCA implementation is available it will be shown and interpreted. Otherwise this report will use the PowerCA solution instead.

5.1.2 Interpretation for results of all Methods

Next, in order to give a first interpretation of the results, the solutions of the second row in Table 2 are displayed in more detail. More specifically, the pairs plot of variable categories for the first 5 dimensions of the manual MCA implementation is shown in Figure 1. This gives a broad overview of the patterns captured by the different dimensions and lets one find pairs of dimensions of further interest. Note that plots of any pair of dimensions for this type of solution must always have an aspect ratio equal to 1. However, in the pairs plot this quickly becomes undesirable as then a common interval of the coordinates for all dimensions needs to be found, often resulting in a rather uninformative pairs plot. It is therefore not implemented in this report but the reader should keep the indicated scales of each dimension in the pairs plot in mind, as indicated in the plot. Individual pairs of dimensions are always plotted with an aspect ratio equal to 1.

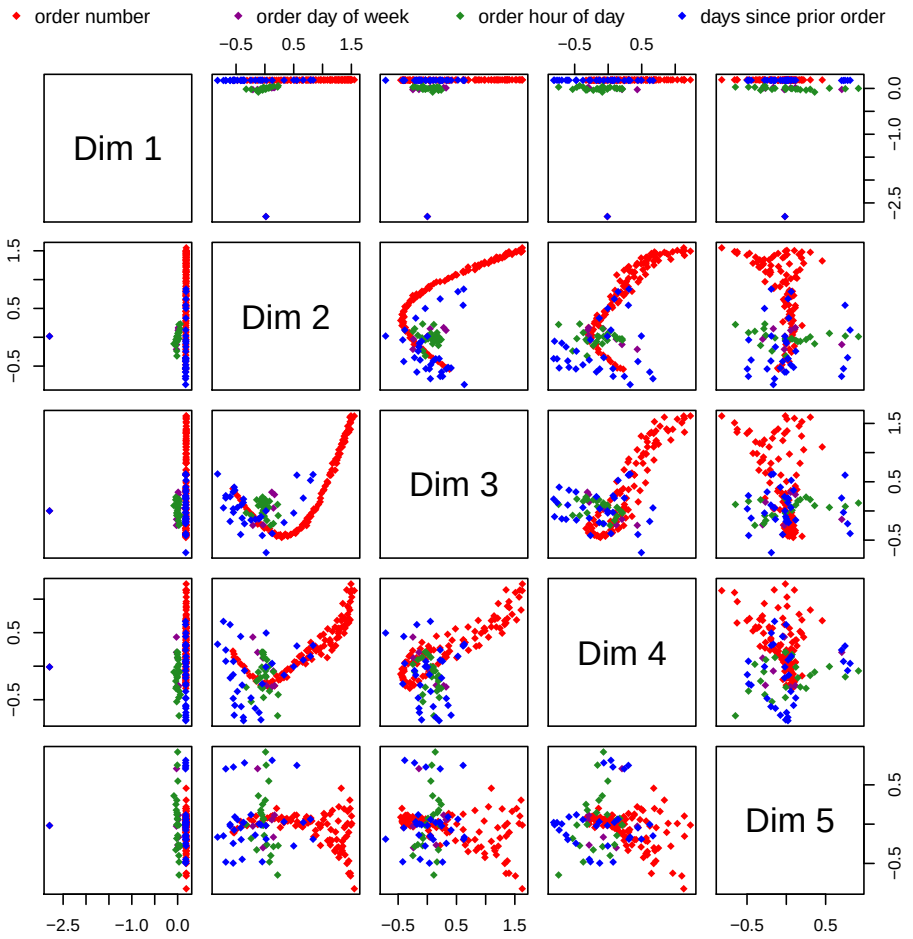

Figure 1: Pairs plot of Manual MCA, 1,000,000 observations

First, it should be mentioned that the first dimension is completely dominated by the association between two values, namely `order_number = 1` and `days_since_prior_order = -1`. Since in the data set these values always occur together, every first order by a user has the value -1 for the number of days since the previous order, this fact is also reflected in the results. Though hard to see in the plots, the first dimension groups these two variable categories clearly together on one side, while all other variable categories are groups together on the other side. Additionally, in the first dimension a few green points of the variable `order_hour_of_day` are located slightly apart from the cluster of the remaining green points, on the right. These are the ones representing times at night (1am - 2am) as well as in the evening (6pm - 7pm) which can be interpreted as first orders being less likely in those time intervals.

Next, one can dive deeper into the results and observe that more interesting patterns are occurring in higher dimensions, namely between dimensions two to four. More specifically, dimensions two and three seem to indicate a clear parabola shape between the variable categories. Hence, Figure 2 displays this plot specifically. Note that the four variables present in this table are colored differently in order to differentiate them, as indicated in the legend of the plot. Further, the variable categories for each variable are connected with a line, starting from the lowest value. This first value also has a different shape, namely a star, while the other points are plotted as diamonds. In this way the variables themselves as well as their values can be visualized in a manner that facilitates interpretation.

The parabola shape is formed by the variable categories of the variable `order_number`. More specifically, following the parabola-like shape from left to right, the categories roughly represent low order numbers and increase to the higher order number on the top right. This seems to indicate that similar overall buying behavior exists for orders that were made subsequently, while along the third dimension a similarity between the first few orders and orders 40 to 60 is found. Logically, orders with a number between roughly 10 and 40, and those above 60 seem to indicate opposite buying behaviors along this dimension. Lastly, the plot indicates associations between variable categories of different variables as well. Namely, the blue categories on the right which are closer to higher order numbers belong to low values of the variable `days_since_prior_order`. Hence, orders with numbers between roughly 40 and 60 seem to occur in very short time intervals of only a few days after each other. In these two dimensions, the variables `order_dow` and `order_hour_of_day` do not seem to indicate strong associations among their categories.

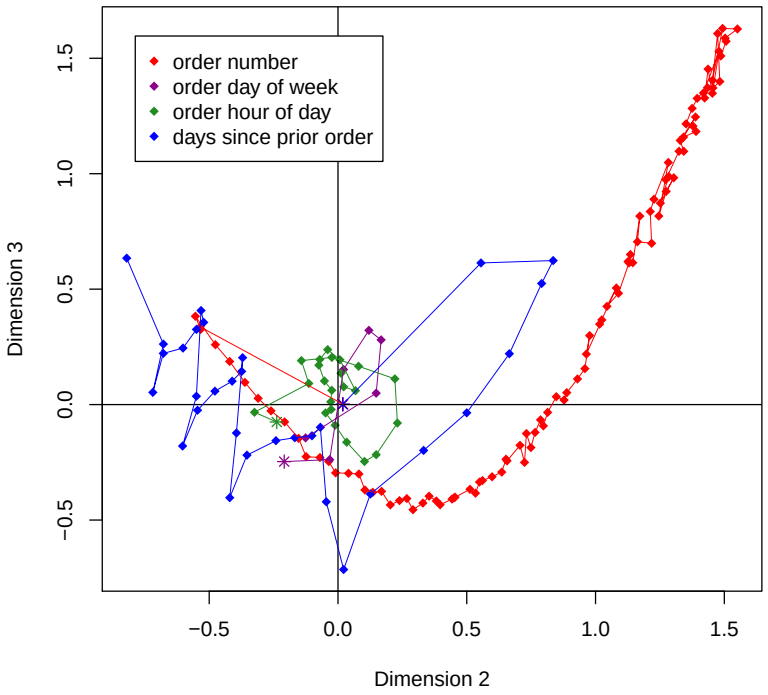

Figure 2: Dimensions 2 and 3 of variable categories, Manual MCA, 1,000,000 observations

5.1.3 Interpretation for Complete Traditional Data

In order to make this first implementation of the given methods on the traditional data table complete, the results of the last row in Table 2 are shown in more detail as well. These include all 3.4 million observations in this table and are hence the most representative. Note that FactoMineR was not able to compute a solution within a reasonable amount of time and hence the result of the manual MCA implementation will be shown.

	Dim 1	**Dim 2**	**Dim 3**	**Dim 4**	**Dim 5**
RMSE of results in second and third row of Table 2	0.006	0.017	0.040	0.884	0.512

Table 4: RMSEs between solutions from Table 2

The pairs plot of the first five dimensions looks relatively similar to the one in the previous section, using only 1 million observations. The first dimension is again dominated by the association of the first order. This indicates that the 1 million observations used in the previous section already included a lot of the information present in this data table. The pairs

plot is hence not displayed a second time but only the RMSEs of the coordinates between the two results for each dimension are displayed in table 4. It can be seen that for the first three dimensions the coordinates of the two solutions only differ marginally. For dimensions four and five the additional observations in this data table have changed the solution to a larger degree. Hence the focus in this subsection will be directed to pairs of dimensions which include at least dimension four.

Figure 3 shows the plots of variable categories of dimensions three and four. The variables are colored in the same way as in Figure 2 and the lines indicate the order of the values, with the lowest number being plotted as a star rather than a diamond symbol. Again, the third dimension shows a similar pattern of the categories of the variable `order_number`, this time plotted on the horizontal axis. The fourth dimension changes this pattern up slightly, indicating a stronger similarity between low order numbers and the ones around 50 (center of the plot where the red points start the circle-like shape to the left and then transfer into a roughly linear shape). Nonetheless, the general pattern of the third dimension with a similar interpretation as given in the previous paragraph persists.

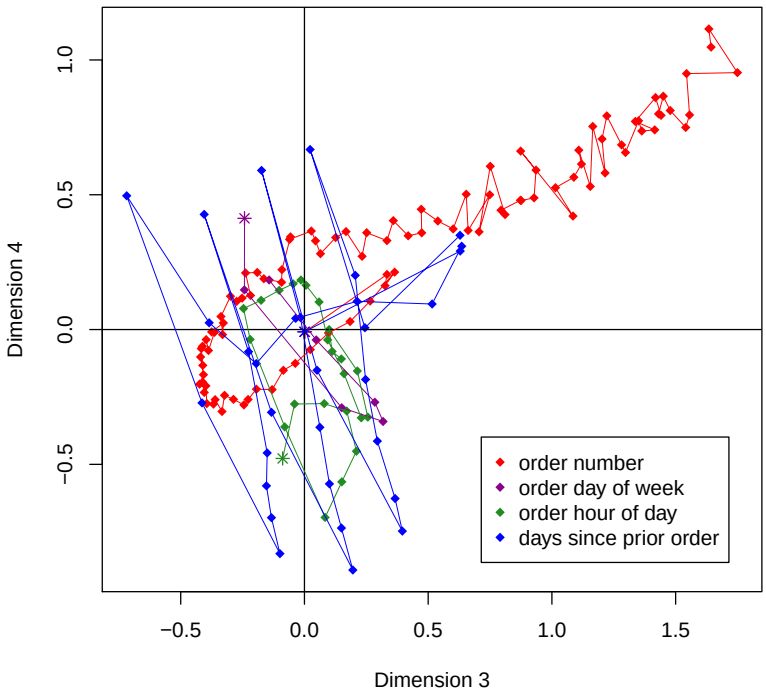

Figure 3: Dimensions 3 and 4 of variable categories, Manual MCA, all observations

Another interesting finding becomes apparent in dimensions three and four. On the left side of

the plot, above the could of points there are four blue and one yellow variable categories which seem to differentiate themselves from the rest. These correspond to the values `order_dow = 0` (which indicates a Sunday) and `days_since_prior_order = 7,14,21,28`. Hence, a clear association between orders that were made on a Sunday and periods in between orders that are multiples of 7 is indicated here. This suggests that a substantial number of customers make repeated purchases on Sundays. Further, the two blue variable categories that are closest to the red one are the two values `days_since_prior_order = 14,21`, which indicates that within this pattern, customers seem to wait for two or three weeks more often in between their orders, than they do one or four weeks. Given the fact that the data set stems from an online grocery store, one can make the interpretation that customers seem to prefer shopping for baskets of groceries which last several weeks on Sundays, comfortably from home when they have the time to do so. For the providers of such services like Instacart, this could indicate that special offers should be directed to these customers on Sundays in order to increase revenue.

The above findings represent a first exploration of the data set, where it has been reduced to its core which allows for a rather traditional implementation of MCA. The Burt matrix remains in reasonable size and hence the methods can run within seconds, independent from the number of observations. One only needs to calculate the Burt matrix from the data table. Hence, for parts of shopping cart data sets that can be reduced to interesting variables which do not amount to an extreme number of total variable categories, Multiple Correspondence Analysis represents a suitable tool for investigating the relationships among them. It might be advisable to implement the algorithm by oneself, rather than relying on a pre-programmed package such as FactoMineR, as these do get significantly slower, the higher the number of observations becomes.

5.2 Data Table containing many categories

Next, the two methods that persisted in the previous section, namely the manual implementation of MCA and PowerCA, are investigated further in terms of their scalability to larger data sets. More specifically, it is of interest to find out how many total variable categories can be handled by the methods in a reasonable amount of time. Again, the methods will first be compared computationally, followed by an analysis and interpretation of the actual result.

5.2.1 All products

In order to test the methods on a data table containing a large number of variable categories, two data tables from the ISCD are agglomerated as described in Section 3.2. Note that the categories of these variables are described in Table 1, Section 3 and that appropriate histograms display the frequencies of the variable categories in Appendix A, Figures 9 - 15. The methods are compared in terms of their running times and the number of observations is gradually increased. Note that largely due to the structure of the variable `product_id`, when taking a subset of the observations one automatically gets a subset of the total variable categories as well. This comes in handy as then the size of the Burt matrix B gradually increases as the number of observations increases. This allows for finding an appropriate threshold for each method in terms of the size of B. To this end, the implementations of the manual implementation of MCA as well as PowerCA have been improved by replacing the matrix calculations with sparse matrices as much as possible. For this, the R packages `matrix` and `sparseinv` have been used. Further, a manual algorithm to compute the indicator matrix as well as the Burt matrix directly in sparse matrix format has been implemented since the built in functions broke down relatively quickly when dealing with many variable categories. The pseudocode of this algorithm can be found in Appendix C. Again, first the running times of the given methods are shown in Table 5 on different amounts of observations in order to give a general overview of their performance.

observations	size of B[7]	**FactoMineR**	**Manual MCA**	**PowerCA**
10,000	3,258	-	3:24	0:07
50,000	9,016	-	-	0:57
100,000	12,831	-	-	2:03
250,000	19,606	-	-	6:13
500,000	25,339	-	-	15:21

Table 5: Running times of the methods on the data table with many categories, in minutes and seconds

FactoMineR was also tried on the smallest subset, but the focus of this subsection is clearly on the other two methods since it can directly be seen that it is not able to handle a large amount of variable categories at all. Note that a method was stopped when it couldn't find a solution within 15 minutes. Further, the manual implementation of MCA, though improved using sparse matrices, is also reaching its limits quickly. The results indicate that for a Burt matrix of size (roughly) a few thousand, the running times are within minutes but start to

[7]Note that B is a square matrix. Hence an entry of 3,258 corresponds to a matrix of size 3,258 $\times$ 3,258.

take much longer for larger B. This is due to the Singular Value Decomposition (SVD) which is part of the MCA algorithm. The size of the Burt matrix B is exactly equal to the size of the standardized residual matrix S on which the SVD is performed (see Section 4.2 for details). It seems that the regular MCA algorithm, without any extra improvements that could overcome the SVD bottleneck, can only compute results within a reasonable amount of time for Burt matrices of size (roughly) a few thousand.,

In contrast to this, the PowerCA algorithm is performing much better. Burt matrices of size up to 20,000 can be implemented within a few minutes. Larger matrices also seem to be possible but start to take longer. This can be explained by the structure and general idea of the PowerCA algorithm. The SVD is circumvented by the use of the so called Power Method to extract the most dominant eigenvector eingenvalue pair from the standardized residual matrix S (see Section 4.3 for details). Hence the running times are much faster.

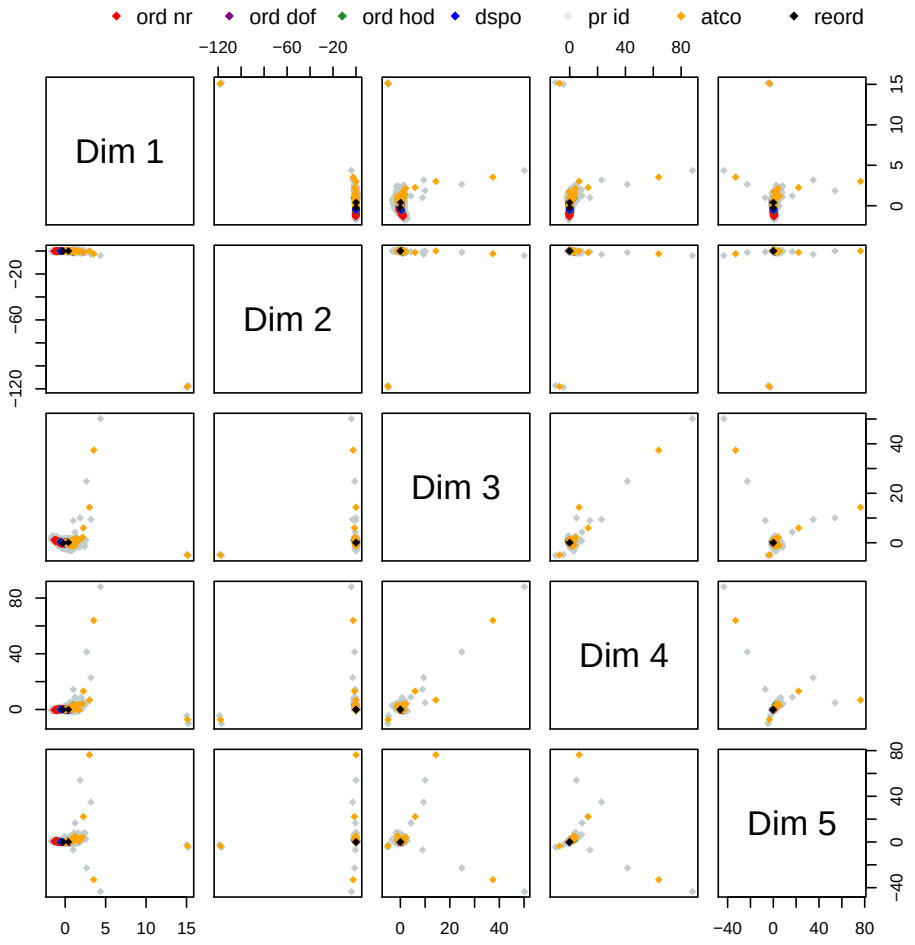

Figure 4: Pairs plot of PowerCA, 500,000 observations

In order to make the exploration of the methods on this first table with a substantial amount of variable categories complete, the actual results of the last row in Table 5 are analyzed in more detail as well. Figure 4 shows the pairs plot of the first five dimensions of the result. Note that in pairs plots with more than four variables, some are abbreviated, as indicated in Table 6 in Appendix B. One can immediately see that generally all variable categories cluster together while just a few differentiate themselves. These are three variable categories that clearly cluster together in both the first and second dimension (see pair Dim1 - Dim2 in Figure 4). These three correspond to `add_to_cart_order = 58` and the two products *Organic Ranch Dressing* and *Knorr Pasta Cheddar Chipotle Sauce Mix*. It seems that in the given data set a strong tendency exists for customers to buy these two products (which are both sauces) as the 58^{th} product in their order, which generally corresponds to rather the end of the shopping trip.

Further, when looking at higher dimensions, more points start to differentiate themselves, more specifically in dimensions three to five. These correspond to the values `add_to_cart_order = 53, 57, 59` and the products *Organic Blueberry Granola Bars*, *Luxury Volume Shampoo*, *Glutenfree Toaster Strudel Blueberry* and *Chrunchy Organic Chickpeas, Korean BBQ Flavor*. All of these specific products seem to be of a common type, namely organic, luxury and high in quality. They seem to all be bought more towards the end of the purchase at positions around 50 to 60. A possible indication of this result for an online grocery store could be to specifically place such products when customers seem to already have about 50 products in their basket and are approaching the end of their purchase. This could be an additional pop-up or special placement just before checking out for these types of products.

It should be noted though that when running MCA on a Burt matrix of size (roughly) 25,000 times 25,000, one gets the same amount of points in the variable categories plots, hence interpretation might generally be difficult. Further, in the given results the other variables like `order_dow`, `order_hour_of_day`, or `reordered` did not differentiate themselves from the large cloud of points. A possible approach to this might be to group some variables with thousands of categories into meaningful groups to make interpretation easier or take a subset in a meaningful way. Nonetheless, the results clearly indicate that PowerCA generally represents a strong method to be used on data sets that stem from online grocery stores. Data Tables having up to around 25,000 total variable categories can be analyzed using PowerCA within a reasonable amount of time. For even larger tables it seems appropriate to assume that the number of total variable categories could be reduced to meaningful groups or a meaningful subset of up to around 25,000, which again makes PowerCA feasible.

5.2.2 Products bought at least 50 times

As indicated previously, given the limits of the methods for the total number of variable categories it makes sense to either group the observations or take a subset in a meaningful way. For the results discussed in this subsection this approach is applied, namely from the data table described in Section 3.2, only orders of products that were bought at least 50 times in total are considered. This implies ignoring a certain amount of information present in the given data table, but this amount turns out to be only 1.3% of the total number of observations. This is due to fact that products bought at least 50 times make up roughly 55% of the total number of products but account for 98.7% of the orders in the given data table.[8] This subset hence directs the focus to the most important products while making the implementation of PowerCA feasible with a total of 27,518 variable categories. The running time for the PowerCA algorithm on the Burt matrix of this data table was 20 minutes and 41.8 seconds.

The pairs plot of the first 5 dimensions of the PowerCA solution on this data table can be found in Figure 5. The first relation that can be identified is that the first dimension again accounts for the the two observations `order_number = 1` and `days_since_prior_order = -1` and clearly separates those two from the cloud of the rest of the points. This is further confirmed by observing that the two ratios $R_1^{(ik)}$ for these two points are by far the largest for this dimension.

Looking into higher dimensions it can be seen that the points of the variable `product_ID` do not form strong distinctive patterns in the first 5 dimensions, but mostly cluster together in a large cloud of points. Further, points of this variable generally do not account for the largest ratios $R_1^{(ik)}$ in these dimensions. Hence, more meaningful results should be expected in the patterns of the other variables. One pattern that strikes out is that of the variable `add_to_cart_order`, plotted in orange. In many of the plots of these dimensions, when considering the points in increasing order for this variable, they begin following somewhat of a polynomial shape roughly in the center of each plot. Then towards the large values of this variable, the points completely diverge from this pattern and seem to be positioned rather uncontrollably. For a more detailed analysis the plot of dimensions 2 and 3 is shown in Figure 6, where this pattern becomes visible as well. Note that for the variable `product_ID` a rather bright color is chosen in order to make the plots readable. Also, this variable is not connected with a line since its points do not follow a particular order.

[8]The full data table from section 3.2 has 33,819,106 observations. When deleting the orders of those products that were bought less than 50 times this number reduces to 33,389,913

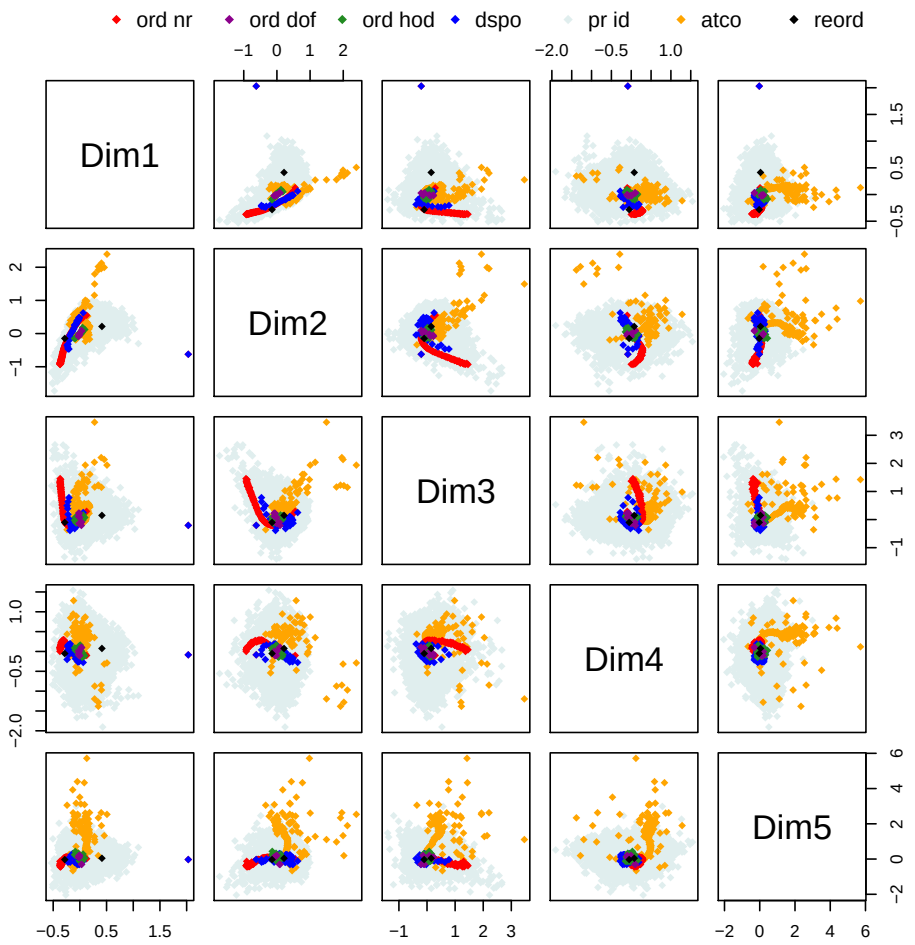

Figure 5: Pairs plot of PowerCA, Products bought at least 50 times

Figure 6 suggest that products that were added last to the shopping cart distinguish themselves strongly from the rest of the products. More specifically, this applies especially to large orders of many products since those are the ones providing observations with large values for `add_to_cart_order`. This diverging pattern begins roughly around the value `add_to_cart_order = 70` indicating somewhat of a threshold. Products that were added roughly after position 70 in the shopping cart show a strong divergence from the rest of the products. This information could be used by an online grocery store by specifically placing products that do not relate to the items already in the cart towards the end of the shopping trip.

Further information that can be read from Figure 6 are the polynomial shape of the variable `order number`, plotted in red with a similar interpretation as mentioned in section 5.1.2 as well as the observations of `days since prior order = 7,14,21,28` which differentiate

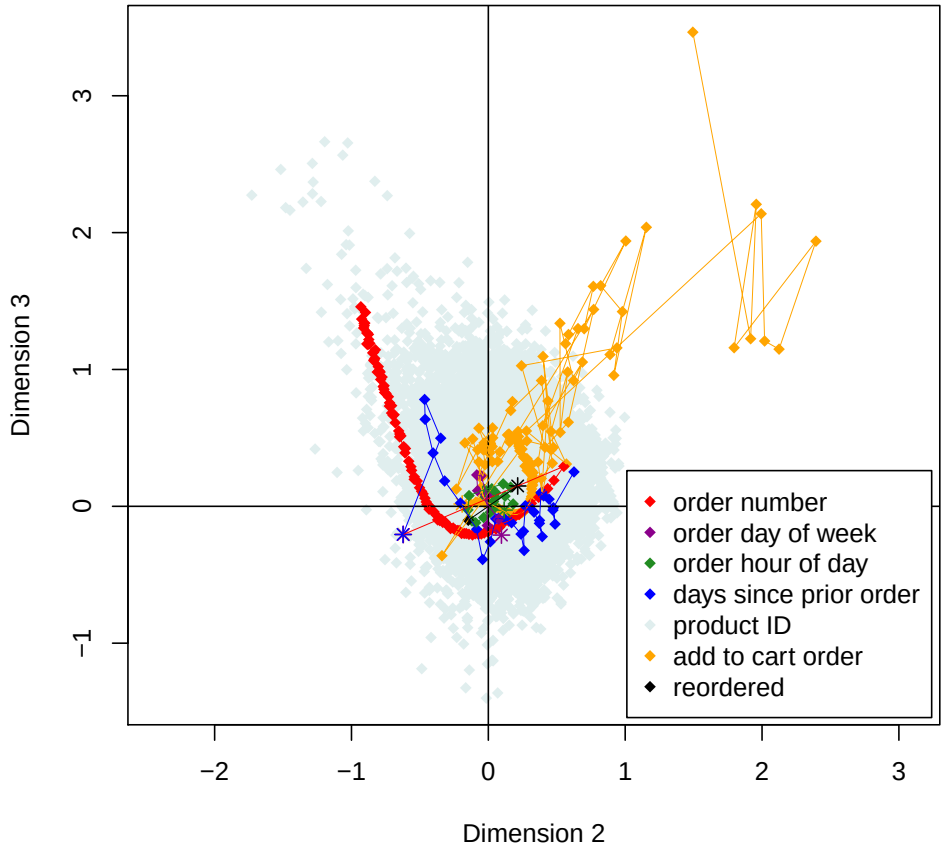

Figure 6: Dimensions 2 and 3 of PowerCA, products bought at least 50 times

themselves as the downward spikes in quadrants III and IV. These again indicate the difference in purchasing behavior on Sundays as described in the same subsection. Interestingly, the point `reordered = 0` is among the 4 points with the largest ratios $R_1^{(ik)}$ for both dimensions 2 and 3. It hence played a dominant part in forming the shape of Figure 6. This can be interpreted as an indication that for products that are added at the end of the purchase, which strongly differentiate themselves from the rest of the points, it is likely that they are bought for the first time.

5.3 ISCD Data Table

As a final step in the analysis and interpretation of results, the data table described in Section 3.3, representing the ISCD as a whole, is used as input for one of the methods. Note that the

categories of the variables in this table are described in Table 1, Section 3 and that appropriate histograms display the frequencies of the variable categories in Appendix A, Figures 9 - 17. Due to the immense size of this data table, PowerCA represents the only feasible method to be applied to it. Further, given the limits for PowerCA found in the previous section for the maximum number of total variable categories, the need for grouping or taking a subset of some of the variables arises again. The given data table contains about 50,000 variable categories which are reduced in the same way as in Section 5.2.2, namely by only considering those products that were bought at least 50 times. This reduces the number of total variable categories to 27,728, making the implementation of PowerCA within a reasonable amount of time feasible. The same reasoning for the choice of this grouping as explained in Section 5.2.2 applies here as well. The running time for the PowerCA algorithm on the Burt matrix of this data table was 21 minutes and 57.4 seconds.

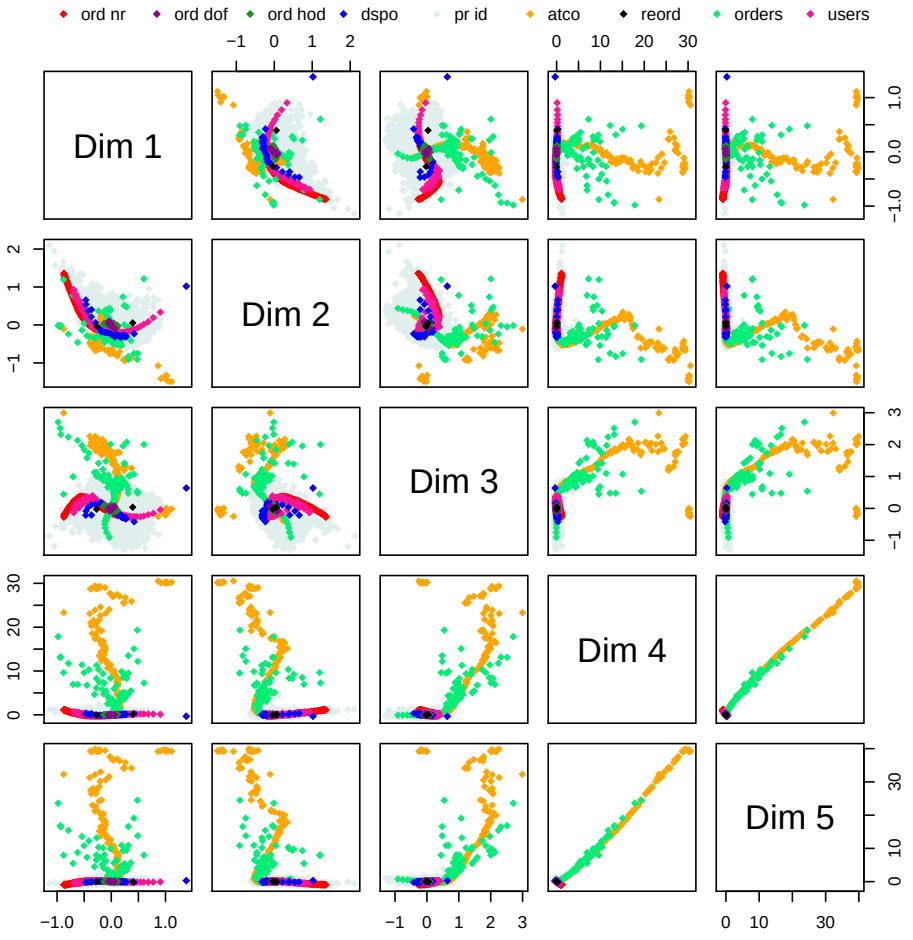

Figure 7: Pairs plot of PowerCA, ISCD Data Table

Figure 7 shows the pairs plot of the first five dimensions of the result of PowerCA on the given

data table. Note that the variables are colored in the same way as in the previous section with the addition of the two variables `users` and `orders`. It can be seen right away that both dimensions 4 and 5 are entirely dominated by the two variables `add_to_cart_order` and `orders`. This is largely due to the construction of the variable `orders`, which groups each order by its size. Hence products that were added to the basket at large positions associate to large order sizes. In all plots of either dimension 4 or 5 with lower dimensions these two variables form a rather straight line in the same direction from low to high values, with the values of `add_to_cart_order` continuing this pattern farther. Note that again the pairs plot does not show individual plots of two dimensions with an aspect ratio equal to 1 due to the same reasons mentioned previously. In this case this leads to a slightly misleading impression of the plots of dimensions 4 and 5 with lower dimensions, but when observing that the scales of these two dimensions are much larger the described pattern becomes clear. Further, the ratios $R_1^{(ik)}$ for the large values of these two variables are highest for both dimension 4 and 5, indicating that they contributed strongly to their shape.

The focus should therefore be directed to the lower dimensions. Here a similar pattern as described in the previous paragraph can be observed for the two variables `order_number` and `users` where their categories, plotted in red and pink, follow almost the same shape from low to high values. This can again be explained by the nature of the variables, namely that high order numbers associate with those users that made comparably many orders in the first place. The intentions described in the last paragraph of Section 3.3 did therefore not realize themselves much, but the similarity within each pair of variables dominated the results.

Nonetheless, a few differences between these pairs of variables can be seen which do have some interpretation. Figure 8 displays all variable categories of dimension 1 and 2 specifically. The variable categories are again connected with lines indicating their values from low to high, with a different symbol for the lowest value. When observing the two variables `orders` and `add_to_cart_order` again, colored in light green and orange, one can see that they start from roughly the center of the plot and initially follow a mostly straight pattern downwards. However, at a given point this straight pattern stops and the points follow a rather messy pattern with many observations jumping back and forth horizontally. This is especially true for the the values of `orders`.

Furthermore, the variable `reordered` shows a clear horizontal segmentation in the center of the plot, along the first dimension. Together with the fact that the values of the ratio $R_1^{(ik)}$ are largest for the two values of this variable in the first dimension, with a substantial gap to the following variable categories, this strongly indicates that the first dimension can be interpreted as segregating products by how likely they are to be reordered. This also explains

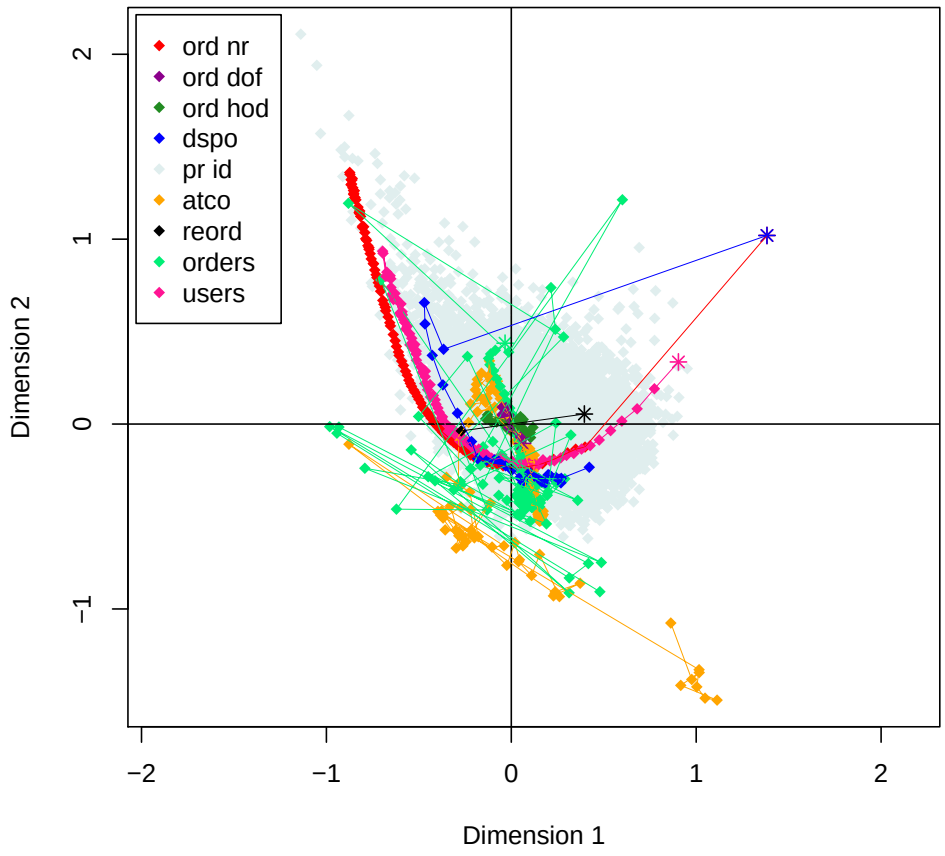

Figure 8: Dimensions 1 and 2 of PowerCA, ISCD Data Table

the horizontal patterns observed in the variables `orders` and `add_to_cart_order`. Large orders with many items seem to either contain mostly reordered products or mostly products bought for the first time. The few points on the bottom left, relating to the highest values of `add_to_cart_order` indicate that the products that were bought at the very end of the largest orders present in the data, were bought for the first time.

6 Conclusion

In this research paper the question of whether Multiple Correspondence Analysis represents a suitable technique in order to visualize large-scale shopping cart data was investigated. More specifically, the intention was to find out at what point MCA would break down due to the computationally expensive Singular Value Decomposition, and what kinds of other techniques

are available to overcome this drawback. In order to answer this question a representative shopping-cart data set from an American online grocery store called Instacart was taken and put into three different agglomerations, where each data table represents a specific use case.

Initially, a data table with a limited amount of total variable categories was created, representing a traditional implementation of MCA via the Burt matrix. Next, more information from the ISCD was included into a second data table, which contains a large amount of variable categories, namely almost 50,000. This data table was used to gradually increase the number of observations and hence the number of total variable categories and showed the computational limits of the given methods. Lastly, a data table representing the ISCD as a whole was created which includes all informative variables of the original shopping-cart data set.

The methods implemented in this analysis are MCA, both as a pre-programmed R package (FactoMineR) and a more efficient manual implementation using sparse matrices, as well as a fast version of MCA, called PowerCA. The computational results showed that all methods can be applied to the traditional data table with a limited amount of variable categories and give a solution within a reasonable amount of time. The limit of FactoMineR was found rather quickly on this data table though in terms of the total number of observations that it could handle. With 1,000,000 observations FactoMineR took just over 7 minutes to find a solution and with all 3.4 million observations in this data table it was not able to compute a solution within 15 minutes. This is due to the fact that FactoMineR takes in the original data table as a whole and computes the Burt matrix by itself as well as many other computational results of MCA, which results in long running times with many observations. The manual MCA implementation as well as PowerCA were able to compute solutions within seconds. Here the Burt matrix had to be calculated separately which took more time for more observations, but the actual algorithms applied to the Burt matrix ran extremely fast. The total amount of variable categories on this data table is equal to 163.

The second data table was used to gradually increase the number of variable categories and find the computational limits of the given methods. FactoMineR did not find a solution within 15 minutes for the smallest subset with 3,258 variable categories. The manual MCA implementation did but broke down in the same time limit for 9,016 total variable categories. Only PowerCA was able to handle much larger amounts of variable categories but showed its computational limits on this data table as well, namely for around 25,000 variable categories with a running time of roughly 15 minutes. Computational results for more variable categories can be found with PowerCA if one is willing to increase the time limit and wait longer for the algorithm to finish, but for this report a time limit of 15 minutes was used.

These limits hence did not give a solution for visualizing the complete data table, as it had roughly twice the number of total variable categories. A solution for this problem is to either group the variable categories or take a subset in a meaningful way in order to reduce the number of total variable categories to roughly 25,000. For the given data table a representative subset was found by only considering products that were bought at least 50 times, resulting in 27,518 total variable categories. PowerCA ran on this subset successfully with a running time of 20 minutes and 41.8 seconds.

Lastly, for the data table representing the ISCD as a whole the same approach for reducing the number of total variable categories had to be applied. Therefore, two additional variables were grouped, namely users by their total number of orders and orders by their size. Additionally, the same subset for the products as taken for the previous data table was used. This resulted in a data table with 27,728 variable categories which included all informative variables of the ISCD. PowerCA ran successfully on this data table in 21 minutes and 57.4 seconds.

The actual results of the methods on all three data tables were presented and interpreted as well. Here it is useful to consider pairs plots for the first five dimensions in order to have a broad overview of the found patterns as well as coloring the different variables. Further, the two ratios given in section 4.2.3 help interpreting specific dimensions as well as judging the representativeness of certain points within them. When a specific pair of dimensions of interest is found it is useful to connect their points with lines from low to high values in order to facilitate interpretation. In this way relevant relationships between the variables can be derived from the results.

Hence one can argue that MCA does represent a suitable method for visualizing shopping-cart data but only to a certian degree, given the computation limits. The algorithm can be drastically improved by the use of sparse matrix packages, but the computational limit can be expected for around 4,000 total variable categories. For data tables falling below this limit, an efficient implementation of MCA represents a more than suitable technique. For data tables with more variable categories, PowerCA can be readily applied and increases the given limit to around 25,000 total variable categories.

Further, it should be noted that though also PowerCA is limited, it remains questionable whether it would be very advantageous to find ways to increase this limit further. The reason for this is that in the final results one will have as many points in the plots as there are variable categories which quickly makes interpretation difficult when this number becomes extremely large. It can therefore be stated that this computational limit does not impose a huge threat to the applicability of PowerCA to shopping-cart data. This is due to the fact that it is reasonable to assume that for data tables with a larger amount of variable categories,

a meaningful grouping or subset of some variables can be taken while preserving the main information within the data table.

Lastly, it should be noted that PowerCA only gives approxmate results, as the Power Method only approximates the most dominant eigenvalue eigenvector pair of the given matrix. Additionally, this approximation error accumulates the higher the dimensions become that are being analyzed. One should hence use an efficient manual MCA implementation in case the given total number of variable categories allows for it, and otherwise refer to PowerCA.

This research was limited in the sense that for the computational limits found for the used methods, the structure of the variables for this shopping cart data set was such that a large number of variable categories was established by a single variable having a much higher number of total categories than the others. Within the maximum amount of variable categories of around 25,000, no appropriate grouping of the variables could be found that resulted in several variables contributing to the large number of variable categories similarly. It could be investigated further to what degree this impacts the computational aspect as well as the interpretation of the results. Finally, the applicability of other, recently invented dimensionality reduction techniques to the specific case of shopping cart data remains an interesting problem where the current limits are likely to give possibility of expansion going forward.

References

Benzécri, Jean-Paul. 1969. "Statistical analysis as a tool to make patterns emerge from data." *Methodologies of Pattern Recognition*, 35–74.

Blattberg, Robert, Byung-Do Kim, and Scott Neslin. 2008. "Market Basket Analysis." *Database Marketing. International Series in Quantitative Marketing* 18: 339–51.

Burt, Cyril. 1950. "The factorial analysis of qualitative data." *British Journal of Statistical Psychology* 3 (3): 166–85.

Di Franco, Giovanni. 2016. "Multiple correspondence analysis: one only or several techniques?" *Quality & Quantity* 50 (3): 1299–1315.

Dietrich, David, Barry Heller, and Beibei Yang. 2015. *Data Science and Big Data Analytics - Discovering, Analyzing, Visualizing and Presenting Data*. John Wiley & Sons.

Eckart, Carl, and Gale Young. 1936. "The Approximation of one Matrox by another of lower rank." *Psychometrika* 1 (3): 211–18.

Ghodsi, Ali. 2006. "Dimensionality Reduction: A Short Tutorial." *Department of Statistics and Actuarial Science*.

Gifi, Albert. 1990. "Multivariate Nonlinear Analysis." Wiley.

Golub, Gene, and Charles Van Loan. 1996. *Matrix Computations*. The Johns Hopkins University Press.

Gower, John, Patrick Groenen, and Michel Van De Velden. 2010. "Area Biplots." *Journal of Computational and Graphical Statistics* 19 (1): 46–61.

Greenacre, Michael. 1988. "Correspondence Analysis of Multivariate Categorical Data by Weighted Least-Squares." *Biometrika* 75 (3): 457–67.

———. 2017. *Correspondence Analysis in Practice*. Third Edit. CRC Press, Taylor & Francis Group.

Greenacre, Michael, and Jörg Blasius. 2006. *Multiple Correspondence Analysis and Related Methods*. CRC Press, Taylor & Francis Group.

Hirschfeld, Herman Otto. 1935. "A connection between correlation and contingency." *Mathematical Proceedings of the Cambridge Philosophical Society* 31 (4): 520–24.

Horst, Paul. 1935. "Measuring Complex Attitudes." *Journal of Social Psychology* 6 (3): 369–74.

Hotelling, Harold. 1933. "Analysis of a complex of statistical variables into Principal Components." *The Journal of Educational Psychology* 24 (6): 417–41.

Husson, François, Julie Josse, Balasubramanian Narasimhan, and Robin Geneviève. 2019. "Imputation of Mixed Data With Multilevel Singular Value Decomposition." *Journal of Computational and Graphical Statistics* 28 (3): 552–66.

Instacart. 2017. "The Instacart Online Grocery Shopping Dataset 2017." https://www.instacart.com/datasets/grocery-shopping-2017.

Iodice D'Enza, Alfonso, Patrick Groenen, and Michel Van de Velden. 2020. "PowerCA: A Fast Iterative Implementation of Correspondence Analysis." *Advanced Studies in Behaviormetrics and Data Science. Behaviormetrics: Quantitative Approaches to Human Behavior* 5: 283–96.

Iskandar, Adi Panca Saputra, Kheri Arionadi Shobirin, and Komang Oka Saputra. 2017. "Analysis of Shopping Cart At Drugs Store By Using An Apriori Algorithm." *International Journal of Engineering and Emerging Technology* 2 (1): 97–103.

Jolliffe, Ian. 2005. "Principal Component Analysis." *Encyclopedia of Statistics in Behavioral Science* 3: 1215–20.

Klema, Virginia, and Alan Laub. 1980. "The Singular Value Decomposition: Its Computation and Some Applications." *IEEE Transactions on Automatic Control* 25 (2): 164–76.

Korobeynikov, Anton, and Rasmus Munk Larsen. 2019. "svd: Interfaces to Various State-of-Art SVD and Eigensolvers." https://cran.r-project.org/web/packages/svd/svd.pdf.

Le Roux, Brigitte, and Henry Rouanet. 2010. *Multiple Correspondence Analysis*.

Markos, Angelos, George Menexes, and Theophilos Papadimitriou. 2009. "Multiple correspondence analysis for "tall" data sets." *Intelligent Data Analysis* 13 (6): 873–85.

Padhi, Ila, Jibitesh Mishra, and Sanjit Kumar Dash. 2012. "Predicting Missing Items in Shopping Cart using Associative Classification Mining." *International Journal of Computer Applications* 50 (14): 7–11.

Saad, Yousef. 2011. *Numerical Methods for large Eigenvalue Problems*. 2nd ed. Society for Industrial; Applied Mathematics.

Stewart, Gilbert. 1993. "On the Early History of the Singular Value Decomposition." *SIAM Review* 35 (4): 551–66.

Van Der Maaten, Laurens, Eric Postma, and Jaap Van Den Herik. 2009. "Dimensionality Reduction: A Comparative Review." *Tilburg University Technical Report, TiCC-*

TR 2009-005. https://doi.org/https://lvdmaaten.github.io/publications/papers/TR_Dimensionality_Reduction_Review_2009.pdf.

Yang, Tzyy-Ching, and Hsiangchu Lai. 2006. "Comparison of product bundling strategies on different online shopping behaviors." *Electronic Commerce Research and Applications* 5 (4): 295–304.

Appendix A - Variable Frequencies

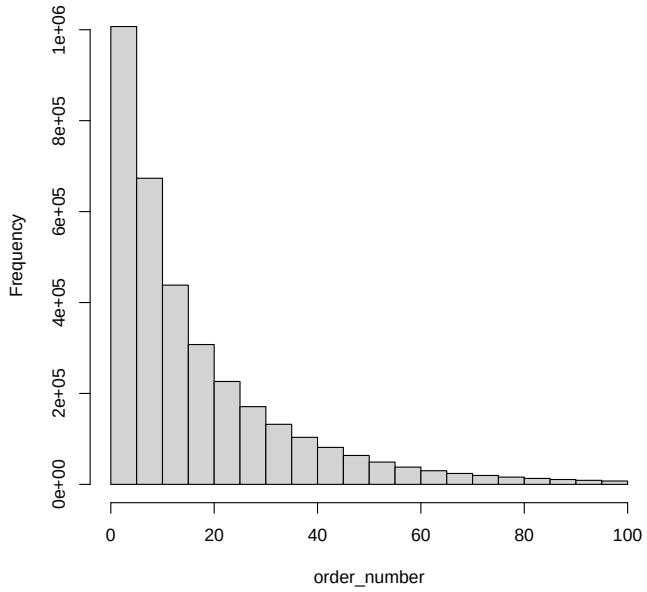

Figure 9: Histogram of the Variable order number

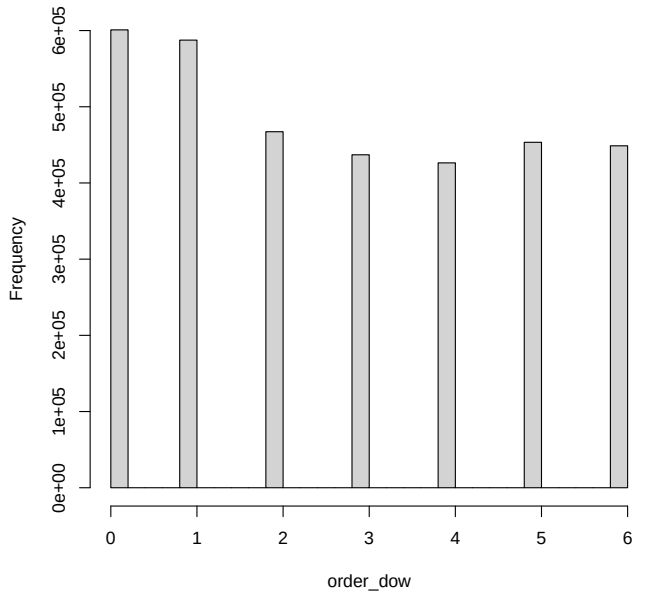

Figure 10: Histogram of the Variable order dow

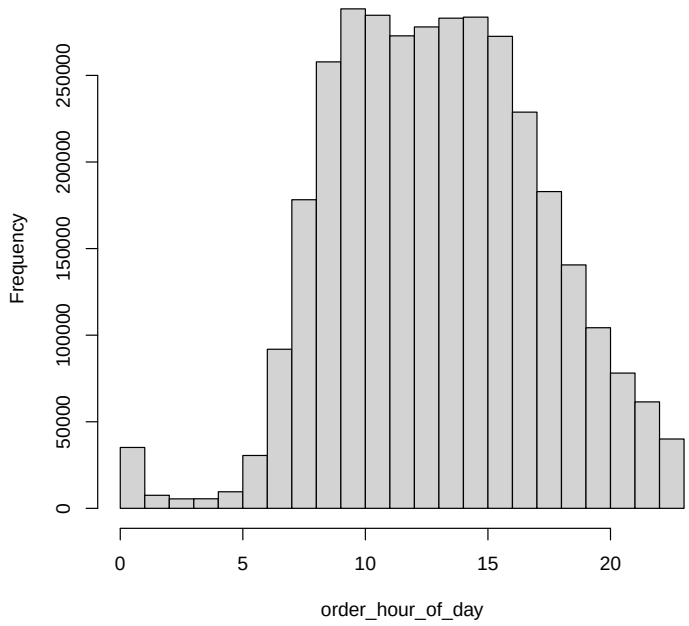

Figure 11: Histogram of the Variable order hour of day

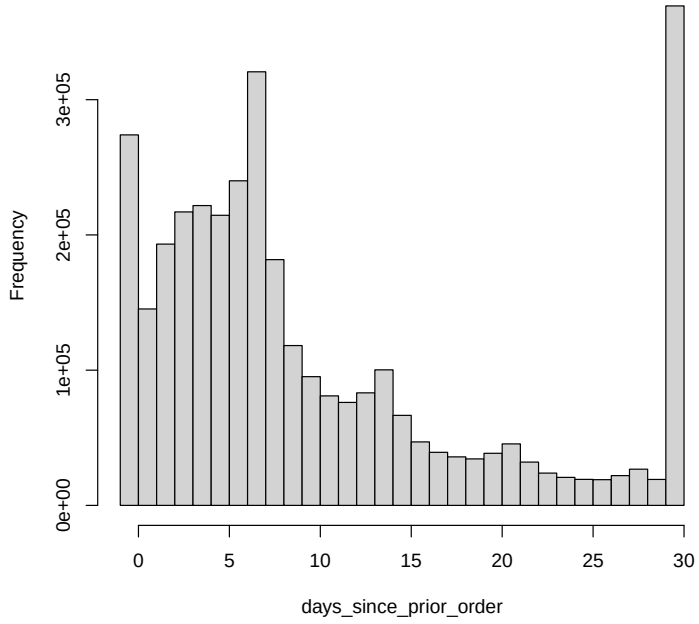

Figure 12: Histogram of the Variable days since prior order

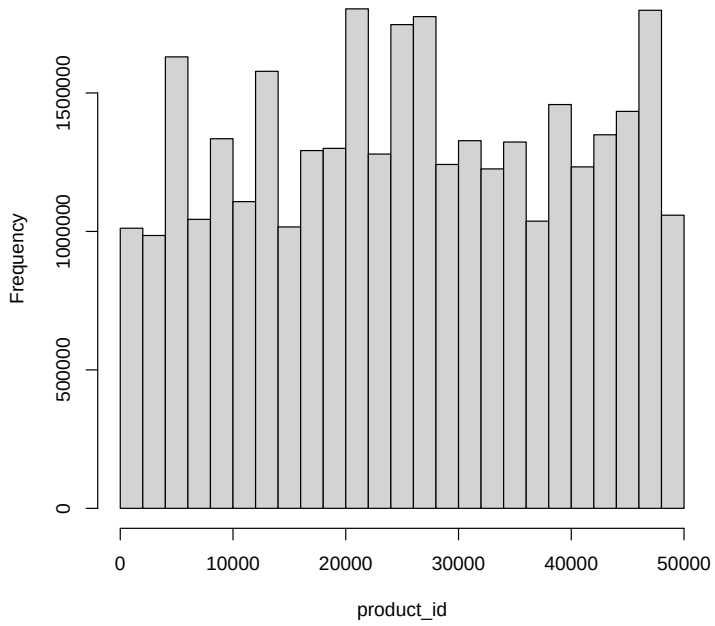

Figure 13: Histogram of the Variable product id

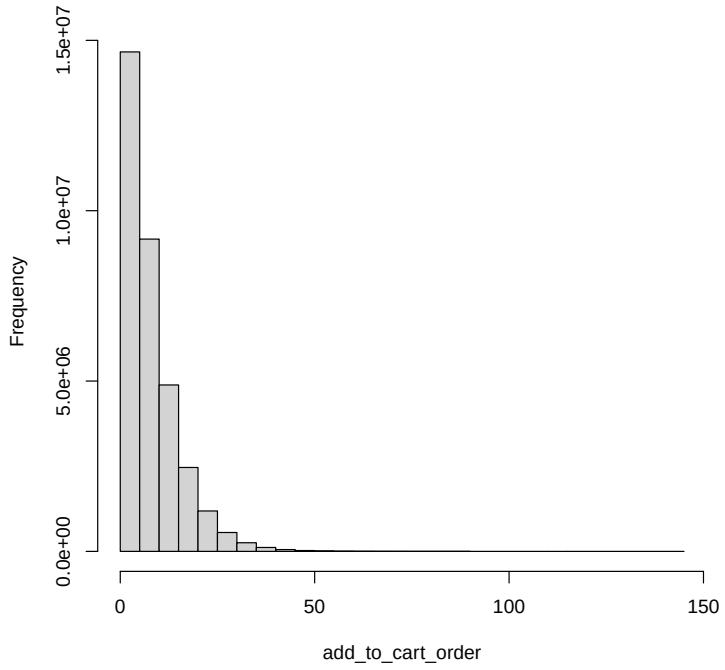

Figure 14: Histogram of the Variable add to cart order

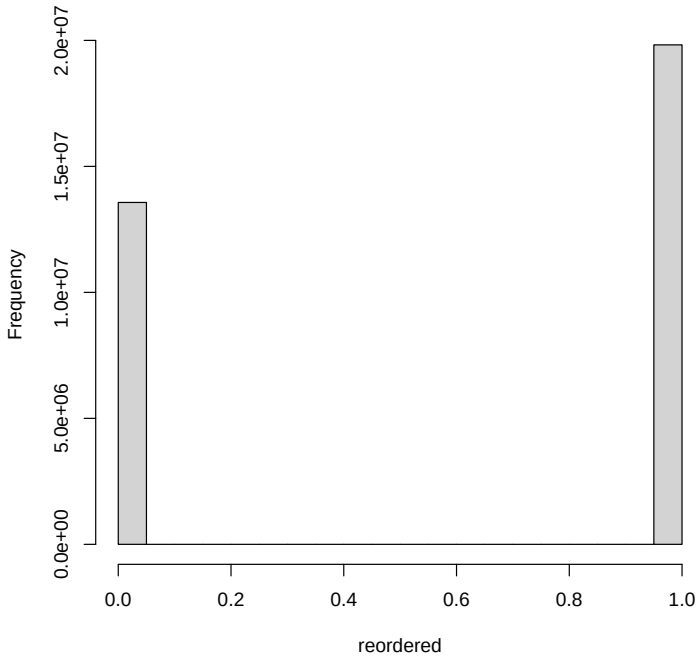

Figure 15: Histogram of the Variable reordered

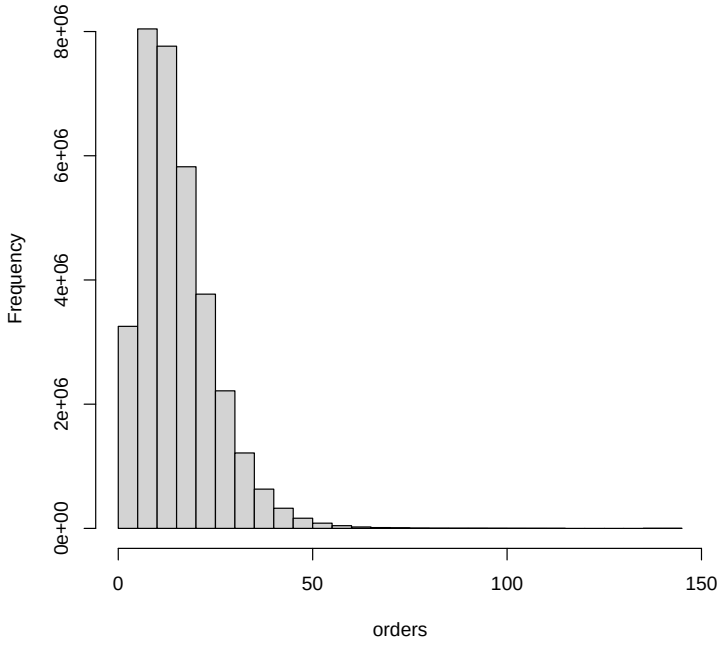

Figure 16: Histogram of the Variable orders

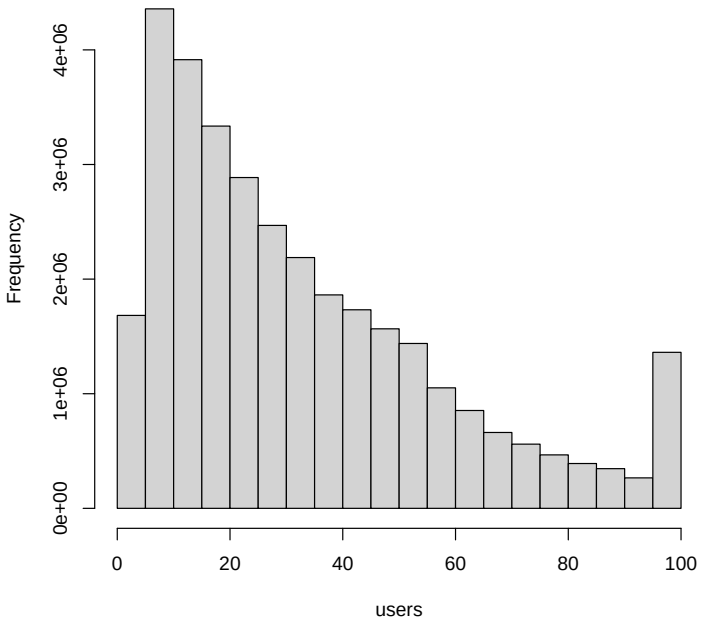

Figure 17: Histogram of the Variable users

Appendix B - Variable Abbreviations

Original Variable	order day of week	order hour of day	days since prior order	product id	add to cart order	reordered
Abbreviation	ord dof	ord hod	dspo	pr id	atco	reord

Table 6: Abbreviation of Variables of Section 5

Appendix C - Pseudocode for calculating G and B

Algorithm 4: Obtain G and B

Input : Data Table with observations as rows and variables as columns, as `data`

Output: Superindicator matrix G and Burt matrix B, as sparse matrices

set a `cursor` equal to 0
set `j_indices` equal to an empty vector
for *column i in* ***`data`*** **do**
    get the unique values of that column as `unq`
    sort those values from low to high
    get the length of that vector as `d`
    map all values of column i from `unq` to those in the sequence `1:d`, save it as `col`
    add to each value in `col` the value of `cursor`
    append `col` to `j_indices`
    add the value `d` to `cursor`
end
get the number of rows in `data` as `r`
get the number of columns in `data` as `c`
set `i_indices` equal to a vector where the sequence `1:r` is repeated `c` times
create the superindicator matrix G as a sparse matrix with entry indices given in `i_indices` and `j_indices`, value = 1
create the Burt matrix as $B = G^T G$
return G, B