NTNU
Norwegian University of
Science and Technology

# Investigating the Potential of Principal Component Analysis on Online Sales Records

**Trym Tarjeison Lekva**

Master of Science in Cybernetics and Robotics
Submission date: June 2018
Supervisor: Harald Martens, ITK

Norwegian University of Science and Technology
Department of Engineering Cybernetics

# Preface

The goal of this thesis is to focus on PCA as a multivariate method to analyze online sales records. Specifically, seasonal and time-of-day effects concerning products and other available information should be investigated. A license was given to the software ”The Unscrambler X” which has been used to produce results and plots. Some parts of the theory section, specifically the NIPALS and fast-food example, was partly taken from my previous works on the subject. As a part of the theis, I have traveled to Macao and visited Associate Professor Yide Liu at MUST (Macau University of Science and Technology). The visit was made possible due the fact that my supervisor, Professor Harald Martens, is appointed as a guest lecturer at MUST. The purpose of the visit was to learn and evaluate the potential of PLS Path Modeling on online sales records, as well as presenting my work-in-progress to Yide. It was through Yide Liu that we got access data records containing online sales of hot pot soups. During my visit, I made a presentation of my work which was sent to the company. As a consequence, the company gave access to more data. I would like to thank Yide Liu for his hospitality during my visit and his efforts in helping me with my work. Unfortunately, PLS path modeling did not become a part of this thesis since its use lies within the field of social sciences and I did not have the time to study it properly. The data was transformed by functions made in Python in order to achieve proper formats for PCA. A special challenge in writing the functions was the handling of Chinese letters. Fortunately, Yide Lui translated the most important parts of the data such that it was easier to understand. In addition to the seasonal and time-of-day effects, the use of PCA in other formats was investigated. While the PCA methods proved to give insights, the conclusions were mainly based on me and Harald Martens investigations on the plots. PCA runs through the veins of Harald Martens, and I realized that it might be hard for outsiders to understand the results. I then chose to test the use of the classification algorithm SIMCA to get more business orientated results, meaning that they are understandable even if one is not familiar with the theory of PCA. I would like to thank Harald Martens for his help and guidance in what I hope is only my first of many steps in multivariate analysis.

# Summary

A multivariate method called principal component analysis has been used to model and analyze patterns in online sales records. Specifically, sales from a Chinese hot pot soup company from 2016 and 2017 have been analyzed. Within the field of data mining, recent literature mentions PCA as a tool for data reduction and fails to comment on its analytic potential [26] [14]. The hot pot sales records have been transformed into three different data structures:

- Daily sales - The products, or the provinces the products were shipped to, was used as columns. The sum of all sales was then put into rows, where each row represents one day.
- Purchase times - The products, or the provinces the products were shipped to, was used as columns. The rows were represented by the total sales within 48 time intervals. The first interval is 00:00-00:30 and the last being 23:30-00:00.
- Customer-product matrix - The products were used as columns. The rows were represented by the different accounts that had bought one or more products, making the dataset represent the product combination each customer has bought.

The daily sales proved to contain interesting information which could be discovered with PCA. A change in purchase behavior over the years and seasonal differences in purchases was discovered. The purchase time format yielded results much similar to if one had summed all the products sold within the time slots. However, one could conclude that different products/provinces did not have a significantly different purchase pattern. This could not have been investigated by solely summing the data. The customer-product matrix proved to be too sparse, consisting of too many zero values, for the use of PCA to be effective. In addition to visual interpretation, a method called SIMCA (soft independent modeling of class analogies) was used. SIMCA is a method for classification, determining if a new sample fits an existing PCA model. A model for the daily sales was built for 2016, and all samples of 2017 were tested on that model using SIMCA. Days in 2017 which were visible as different from 2016 on the model which contained data from both year was selected and used for comparison. The results showed that the SIMCA classified many of the manually selected days as unfit for the PCA model. This indicates that SIMCA could be used to continuously track purchase

behavior and function as some alarm-system for when changes in purchase behavior appear.

# Sammendrag

En multivariat analysemetode som heter Principal Component Analysis (PCA) har blitt brukt til å modellere og finne mønster i salgshistorikk. Salgshistorikken omfatter salg av hot pot supper, solgt i en kinesisk netthandel under 2016 og 2017. Nye bøker innen fagfeltet *data mining* betegner PCA som en datareduksjonsmetode og nevner ingenting angående dens analytiske potensial [26] [14]. Salgshistorikken ble transformert til tre forskjellige datastrukturer som har blitt undersøkt:

- Daglige salg - Kolonnene er utgjort av produkter eller provinsen de ble sendt sendt til. Radene er dager. Èn rute inneholder da f.eks antall produkter solgt i løpet av en dag.
- Salgstidspunkt - Kolennene er utgjort av produkter eller provnsen de ble sendt til. Radene utgjør tidsintervaller fra 00:00-00:30 til 23:30-00:00. Èn rute inneholder da f.eks antall produkter solgt i et intervall over to år.
- Produkt-kundematrise. Kolonnene er utgjort av produkter, radene av kunder. En rad viser da alle produktene èn kunde har kjøpt total over to år.

Ved å undersøke PCA av daglige salg ble flere interessante sammenhenger oppdaget. Sesongforskjeller og endringer i kjøpemønster over perioden på to år ble avdekket. Mønsterene var klarere når produkter ble brukt som kolonner enn når provinsene ble brukt. Salgstidspunkt gav mye av de samme resultatene man hadde fått dersom man hadde summet alle salg i de forskjellige tidsintervallene. Det var lite forskjell mellom de forskjellige produktene, og man kan da konkludere med at forksjellige produkter ikke blir kjøpt på forskjellige tidspunkt. Denne informasjonen får man ikke ved å summe de forskjellige salgene. Produkt-kundematrisen viste seg å ha for mange nullverdier til at PCA kunne modellere den ordentlig. Kundene ble separert på hvor mange forskjellige kjøp de hadde utført i nettbutikken, men i alle tilfeller ble lite av variansen fanget. SIMCA ble brukt på en model laget av daglige salg fra 2016. Dagene fra 2017 ble projisert på modellen fra 2016. En modell bygget på dager fra både 2016 og 2017 viste et klart mønster hvor det på slutten av 2017 ble en endring i kjøpemønsteret. Disse dagene ble manuelt plukket ut og sammenlignet med hvilke dager SIMCA markerte som upassende for modellen bygget på 2016. Det var en stor enighet blant SIMCA og manuelt utplukkede dager. Dette resultatet indikerer at PCA modeller kan bli brukt for å monitorere data i sanntid.

# Table of Contents

# List of Tables

# List of Figures

# Abbreviations

| Symbol | = | definition |
|---|---|---|
| PCA | = | Principal Component Analysis |
| PC | = | Principal Component |
| PLS | = | Partial Least Squares |
| NIPALS | = | Nonlinear Iterative Partial Least Squares |
| SVD | = | Singular Value Decomposition |
| SIMCA | = | Soft Independent Modeling of Class Analogy |

Chapter 1

# Introduction

## 1.1 Problem description

”Multivariate analysis of online sales records. What potential does PCA have in identifying patterns and modeling online sales, and how is it used today?”.

## 1.2 Principal Components

### 1.2.1 What is a principal component?

A principal component is a common direction of variance within a dataset. PCA is a method for analyzing multivariate data by examining the common variances and has found use in several different fields. Large multivariate data sets can be noisy, intimidating and difficult to interpret. PCA projects mean-centered data ($\boldsymbol{X}$) consisting of variables (columns) and samples (rows) onto a new plane. The new plane is represented by scores ($\boldsymbol{T}$) and loadings ($\boldsymbol{P}$). $\boldsymbol{E}$ is the notation of the data not explained by the model, the residuals. Extracting common variances, or patterns, in the data gives several advantages. One can reduce the number of variables needed to represent the same information. This reduction is useful both for visualization and interpretation, as well as compression. For variable selection, it is interesting to see which variables are best explained by this method. Some variables will be more significant than others, why? A few samples might be outliers, meaning that they do not fit the pattern seen in most of the other samples. Have someone made a mistake in the sampling process or is it a reason for this behavior? The model equation of PCA is as follows:

$$\boldsymbol{X} = \boldsymbol{T}\boldsymbol{P}^{\boldsymbol{T}} + \boldsymbol{E} \tag{1.1}$$

### 1.2.2 Early history and usage of principal component analysis

PCA has applications in several fields such as multivariate modeling, data reduction, classification and detection of outliers. Several scientists throughout time have discovered the method. A book written by S. Wold, K. Esbensen and P. Geladi in 1987 [28] covers the early history of PCA. The first discovery of this method was made by K. Pearson in 1901 when he argued that ”it is desirable to represent a system of points in plane, three, or higher dimensioned space by the ”best-fitting” straight line or plane” [11]. Later the method came up in the agricultural studies of Fisher and Mackenzie in 1923 [10]. Fisher and Mackenzie outlined the NIPALS-algorithm, an iterative solution to computing the scores and loadings, which was later rediscovered by the father of S.Wold, H. Wold, in 1966 [27]. In 1933, Hotelling developed the method further, to its present stage [17]. PCA is known as singular-value decomposition in numerical analysis, Karhunen-Loève expansion in electrical engineering and Hotelling transform in image analysis. The rediscovery and different names might have been a drawback for PCA due to the lack of standard terminology, but the method still stands strong today over 100 years after its discovery.

## 1.3 Online sales data and data mining

Online stores have seen, and are continuing to see, a rapid growth [19]. Selling products online provides benefits both to buyer and seller by, e.g., removing geographical restrictions and costs of maintaining a physical store. It also opens up for easier access to customer behavior and purchase patterns. Tracking customer behavior in a physical store requires some sort of loyalty card or way of linking purchases to the specific customers. This availability has come hand in hand with a new computer science and statistical field which goes under the name ”data mining.” Data mining is the extraction and presentation of patterns in large datasets through data processing and statistical methods. An example of this is how a group of scientists was able to ”... accurately estimate the current level of weekly influenza activity in each region of the United States, with a reporting lag of about one day” using only Google’s search history [12]. Another story of data mining revolves around the unexpected co-variance between beer and diapers [24]. One would not initially think beers and diapers have a lot in common. In the book *Digital Forretningsforståelse* the author brings up beer and diapers in chapter 17 [16] which is dedicated to data mining and it's increasing importance in business development. Other examples mentioned in the book is how credit companies can predict a coming divorce using data mining [6] and how banks use data mining to expose terrorists [21].

### 1.3.1 Data mining and principal components

So what role does PCA have within the field of data mining? *Data mining: concepts and techniques* [14], a popular textbook on the subject, describes PCA mainly as mean of data reduction and pre-processing. They do not discuss the attributes of scores and loadings, thus removing the analytical aspect of the method. Another data mining book that describes PCA is *Data Mining: Practical machine learning tools and techniques* [26]. This book discusses PCA in chapter 7 *Transformations: Engineering the input and output*. It explains PCA in a clear and conceptual good way, however, neither this book covers essential concepts such as scores and loadings. These are the most cited books when searching "data mining" on Google Scholar. When reading these favorite books on the subject, it seems like PCA role is limited to applications within data reduction.

## 1.4 Hot pot and the principal components of Chinese cuisine

A collaboration between Professor Harald Martens and Associate Professor Yide Liu MUST (Macao University of Science and Technology) made it possible for me to travel and visit Yide in Macau. In advance, Yide had provided sales records from a company which sells different types of hot pot on a Chinese business-to-customer market called JD. According to their website [18] "JD.com is China's largest online retailer and its biggest overall retailer, as well as the country's biggest Internet company by revenue." The sales data contains several metadata, such as shipping province and on what device the purchase was made. This makes it suitable for utilizing techniques within data mining as well as PCA. During the visit in Macao, which took place in March, the results were presented to Yide and some of his students. The results were also forwarded to the company which resulted in them giving access to even more of their sales data. In total, the data contains sales of 47059 products made by 22147 customers. The principal component analysis will be used not only as a mean of data reduction but as an investigation into the product portfolio and its behavior. The analytic potential of principal component analysis will be discussed together with methods suggesting how to get the most out of the models.

Chapter 2

# Theory

The PCA section in this chapter is based on the book Multivariate Analysis of Quality written by Martens & Martens[15] and documents written by CAMO Software [8] [9] [7]. CAMO Software it the company behind The Unscrambler X, a software for easily doing PCA and other multivariate analysis. The Unscrambler X is used for analysis and plot generation in this thesis.

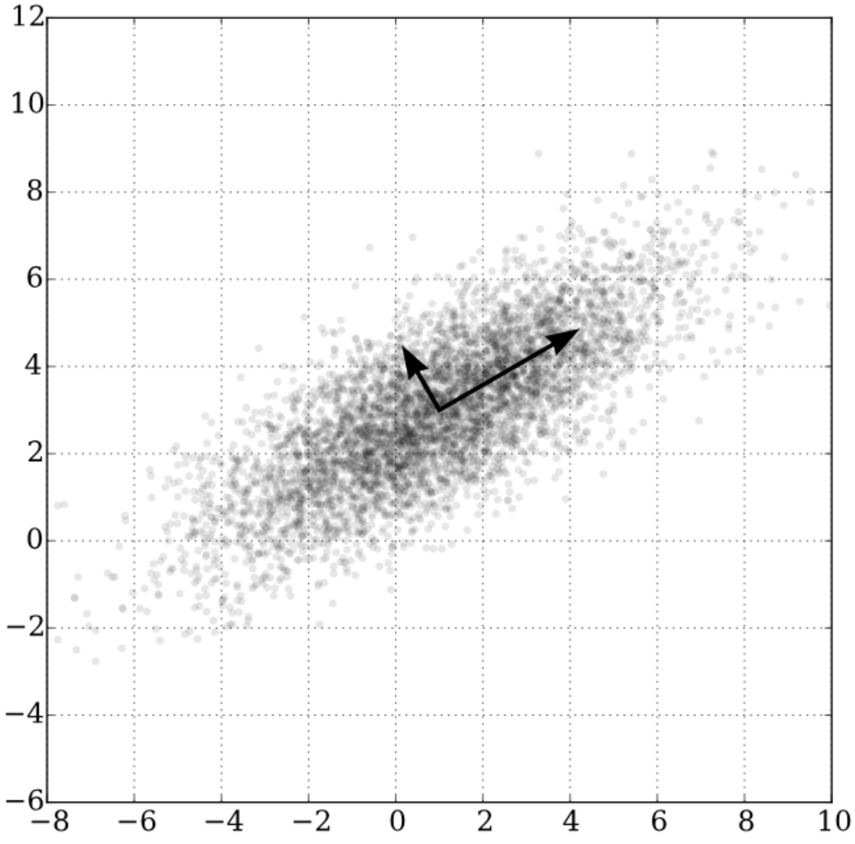
## 2.1 Principal Components Analysis

In this section, principal component analysis will be explained first in a conceptual manner, followed by an example provided by CAMO Software. A commonly used algorithm for calculating PCA, NIPALS, will be presented at the end.

### 2.1.1 What is PCA?

PCA is a method which breaks down data into principal components. In figure 2.1 a two-dimensional example can be seen. The black arrows represent the two principal components which can be found in the two-dimensional plot. The first principal component is the longest arrow, and one can see that this arrow lies parallel to the biggest direction of variance (spread) in the plot. The second principal component lies at a right angle to the first principal component. Geometrically PCA is an orthogonal transform where each axis is at the right angle to the others. Conceptually one can view PCA as a transform which projects the original data onto its preferred coordinate system. An illustration mentioned in [26] explains this well. Imagine that figure 2.1 represents a map of a star cluster that is visible in the sky. The map is made up of properties familiar to the person that drew it. The two directions, with positive and negative values, can represent north, east, south, and west with some common constant to decide the position on the axes. It is clear that the stars are not familiar with the convention made by the artist, and seem to follow some other pattern. The coordinate system consisting of the principal components does not care about concepts (such as north and south), but variance. Maybe one would conclude that the first principal component is the consequence of a large gravitational pull made by, e.g., a black hole, while the second one is made by a smaller one? The loading tells how much each variable contributed, while the score tells each samples placement on the new coordinate system. In figure 2.1 the first component consists of, made by eye measurement, 70% of the horizontal axis and 30% of the vertical axis and vice-versa for the second component. In the possible setting of stars and black holes, loadings show the importance of black holes and scores how much each star are affected by which.

**Figure 2.1** Principal components of bivariate Gaussian distribution [22]

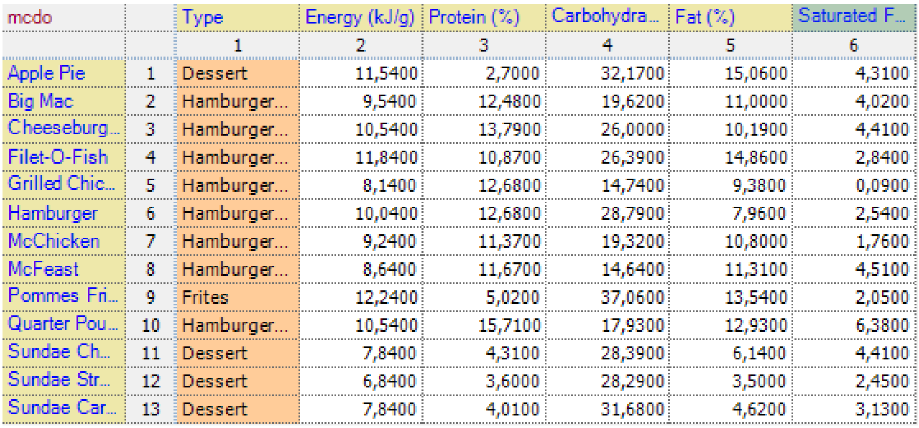


### 2.1.2 The principal components of fast-food meals

An example provided by CAMO Software, which contains nutrition data from Mac Donald's, will be used to explain the steps of PCA. In figure 2.2 the raw data can be examined.

This dataset contains nutrition data from various foods one can buy at a fast-food restaurant. When only looking at the nutritional contents it is difficult to separate be-

**Figure 2.2** Nutrition Data Sample provided by CAMO Software



| mcdo | | Type | Energy (kJ/g) | Protein (%) | Carbohydra... | Fat (%) | Saturated F... |
|---|---|---|---|---|---|---|---|
| | | 1 | 2 | 3 | 4 | 5 | 6 |
| Apple Pie | 1 | Dessert | 11,5400 | 2,7000 | 32,1700 | 15,0600 | 4,3100 |
| Big Mac | 2 | Hamburger... | 9,5400 | 12,4800 | 19,6200 | 11,0000 | 4,0200 |
| Cheeseburg... | 3 | Hamburger... | 10,5400 | 13,7900 | 26,0000 | 10,1900 | 4,4100 |
| Filet-O-Fish | 4 | Hamburger... | 11,8400 | 10,8700 | 26,3900 | 14,8600 | 2,8400 |
| Grilled Chic... | 5 | Hamburger... | 8,1400 | 12,6800 | 14,7400 | 9,3800 | 0,0900 |
| Hamburger | 6 | Hamburger... | 10,0400 | 12,6800 | 28,7900 | 7,9600 | 2,5400 |
| McChicken | 7 | Hamburger... | 9,2400 | 11,3700 | 19,3200 | 10,8000 | 1,7600 |
| McFeast | 8 | Hamburger... | 8,6400 | 11,6700 | 14,6400 | 11,3100 | 4,5100 |
| Pommes Fri... | 9 | Frites | 12,2400 | 5,0200 | 37,0600 | 13,5400 | 2,0500 |
| Quarter Pou... | 10 | Hamburger... | 10,5400 | 15,7100 | 17,9300 | 12,9300 | 6,3800 |
| Sundae Ch... | 11 | Dessert | 7,8400 | 4,3100 | 28,3900 | 6,1400 | 4,4100 |
| Sundae Str... | 12 | Dessert | 6,8400 | 3,6000 | 28,2900 | 3,5000 | 2,4500 |
| Sundae Car... | 13 | Dessert | 7,8400 | 4,0100 | 31,6800 | 4,6200 | 3,1300 |

tween the different types of meals. Let's begin by taking a look at how much variance is explained by each principal component.

### Explained Variance

**Table 2.1:** Explained Variance - Nutrition data

| | PC-0 | PC-1 | PC-2 | PC-3 | PC-4 | PC-5 |
|---|---|---|---|---|---|---|
| Calibration | 0 | 70,70 | 90,29 | 97,33 | 99,99 | 100 |
| Validation | 0 | 53,98 | 77,65 | 86,88 | 99,99 | NaN |

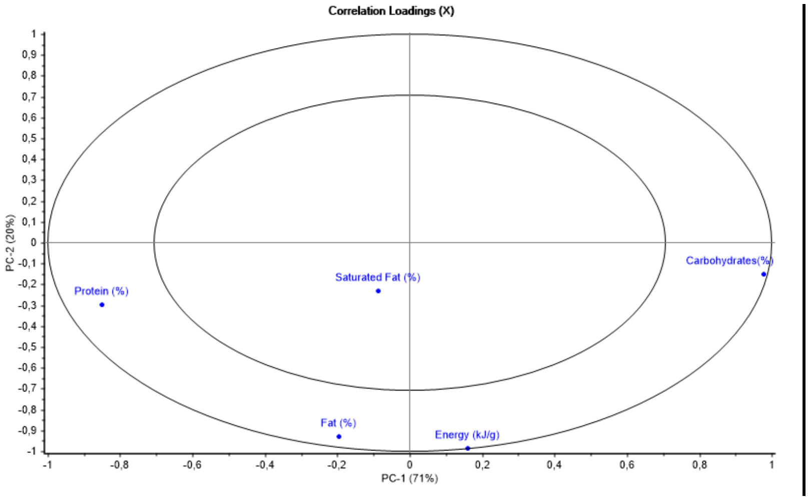
In table 2.1 one can see that the first two principal components capture $90,2923\%$ of the variance in the calibration and $77,65\%$ in the validation. The difference between the calibration and validation is essential. The calibration includes all data samples when building and testing the model. Testing the model upon the data it was made from is cheating, and does not reflect how well it would model new samples. In the validation, one or more samples are hidden from the construction of the model and used for testing. There are several ways of validating models. One way is to hide a subset of the samples, often called a test matrix, during the construction of the model. The test matrix is then exclusively used for validating the model. This method works well when one has many samples, but when one has fewer samples, one can use a technique called cross-validation. In cross-validation, one divides the data into several segments, and then build equally many models in which a new segment is hidden and used for testing. E.g., with a 5% cross-validation, the model is built a total of 20 times, each time with a new 5% hidden from the model. The samples hidden can be a random subset of the dataset. The validation will almost always yield a lower explained variance but at the same time a more realistic indication of how well one would model new variables. Equations 2.1 and 2.2 shows how the explained variance for each component, a, is calculated by dividing the remaining residuals by the total. $\boldsymbol{T_a}$ and $\boldsymbol{P_a}$ represents the scores and loading values which include all principal components from 0 to $a$.

$$Res(a) = \sqrt{\boldsymbol{X^2} - (\boldsymbol{T_a P_a})^2} \tag{2.1}$$

$$\Delta ExpVar(a) = \frac{Res(a-1) - Res(a)}{Res(0)} \tag{2.2}$$

### Loadings

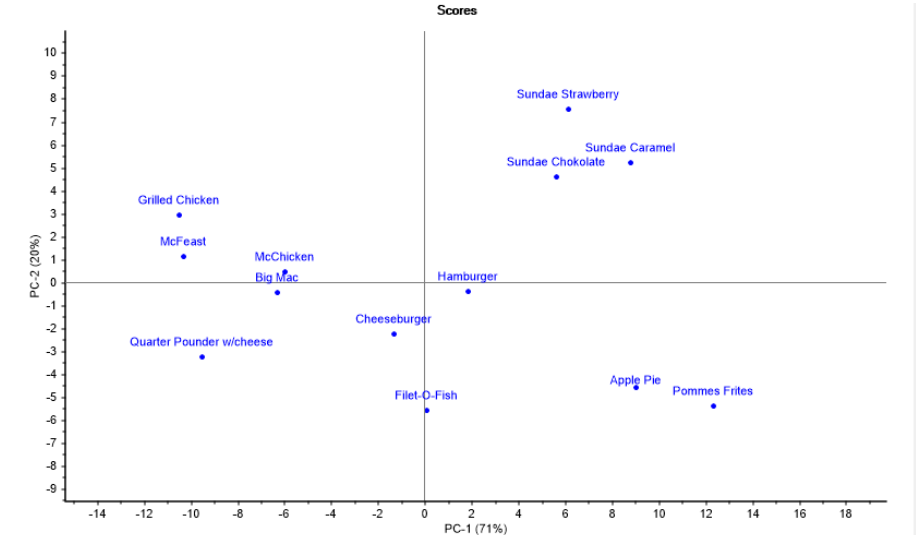
**Figure 2.3** Correlation loadings of nutrition data



In figure 2.3 the correlation loadings of the nutrition data 2.2 is plotted. Note that this plot does not show the actual values of the loadings, but how much the different components explain each variable. Also, the axes are noted by which component they represent followed by a percentage. This is the explained variance from the calibration, not the validation. The negative correlation between protein and carbohydrates is the biggest factors in PC1. For PC2 it is the positive correlation between fat and energy. A high protein source, like meat, does not contain many carbohydrates and there is much energy in fat. The nutrition within the outer ellipse is explained 50-100% (where the outer line is 100%) by the model [25]. This representation makes it quicker to examine which variables are significant in our PCA model. For instance, one can see here that the amount of saturated fat is not an important variable in explaining the two first principal components.

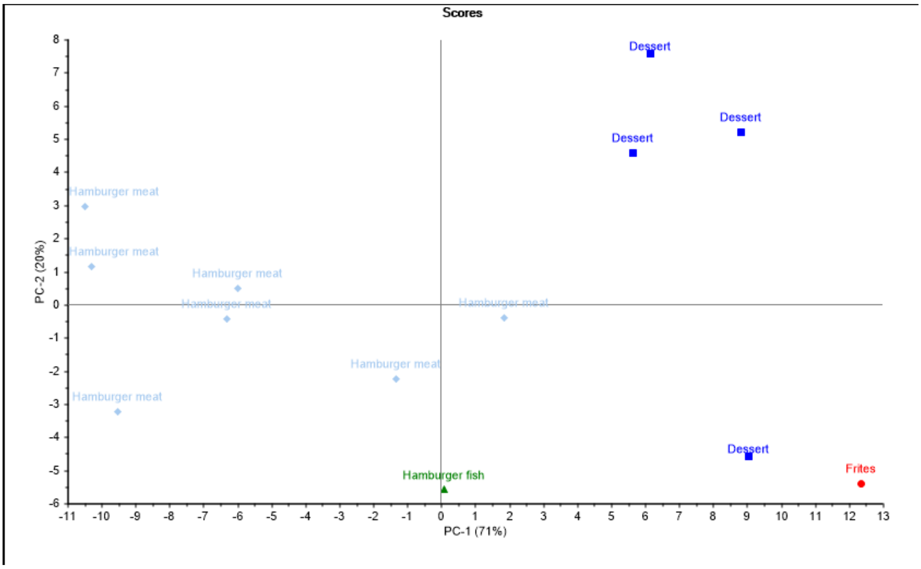
## Scores

**Figure 2.4** Scores plot of meal types. PC-1 and PC-2



The scores tell us how each sample placed upon the new coordinate system. The axes are, as the correlation loadings, marked with which component they represent. As with the loadings, the percentage comes from that component's explained variance in the calibration. If one looks at the location of the different types of meals on the plot seen in figure 2.4, one can see a pattern where the burgers seem to be gathering to the left of PC1. A quick look at the correlation loading plot in figure 2.3 tells us this is the direction mostly influenced by protein. In the top right quadrant lies the Sundae desserts. The top right is the direction in PC1 mostly influenced by a high content of carbohydrates and the direction in PC2 negatively correlated with fat and energy. Figure 2.5 shows the grouping of the meals.

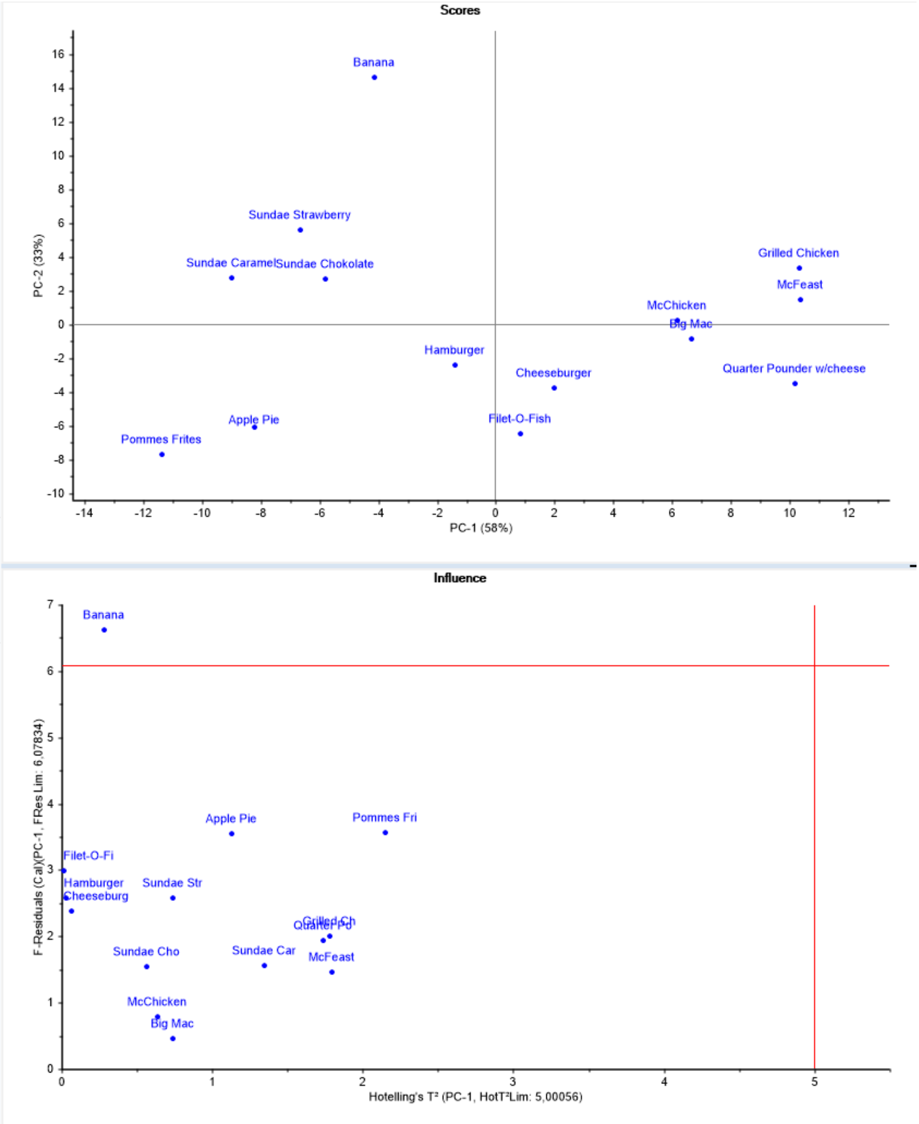
**Figure 2.5** Scores labeled with classification instead of type. PC-1 and PC-2.



What about the other principal components? From figure 2.1 one can see that the explained variance of the data increases for each component one adds. A part of PCA is visual interpretation and reasoning. Do adding more components add any structural information to the analysis, or does it just model noise? Noise can, i.e., be small variations caused by measurement errors or similar. PCA is in many ways a compromise between simplicity and losing the ability to describe the data fully.

### Influence plot

**Figure 2.6** Score and influence plot with nutrition of a banana added to the data. PC-1 and PC-2.



To illustrate the use of the influence plot nutritional data of a banana [3] is added to the dataset. In the figure 2.6 one can see that among the scores, the banana stands alone by itself. However, it is not before examining the influence plot one can tell if there is an outlier in the data. One can see that the nutritional data for the banana falls outside one of the red lines in the influence plot. The two red lines represent two different types of outliers. The horizontal line is the critical limit of each samples residuals based on an ad-hoc rule implemented by CAMO Software. The vertical line shows the hotelling $T^2$ critical limit on an assumption of a student-t distribution [1]. Samples that do not fit the model will be visible outside the horizontal red line. Samples that lie outside the vertical line is considered extremes that fit the model. Such samples can be dangerous for the model as they could give false importance to the variables of which it consists.

**Putting it all together**

The mathematical representation of the PCA of the mean-centered data $\boldsymbol{X}$ is the following:

$$\boldsymbol{X} = \boldsymbol{T}\boldsymbol{P}^T + \boldsymbol{E} \tag{2.3}$$

Where $T$ is the scores, $P$ the loadings and $E$ the residuals which are not explained by the model.

### 2.1.3 The NIPALS algorithm

There are several ways to calculate the scores and loadings of PCA. One of them is called NIPALS. The NIPALS algorithm is an iterative algorithm which was developed by Herman Wold [28]. The NIPALS algorithm shown here is from a compendium for the courses TKJ4175/KJ8175 at NTNU written by Bjørn K. Alsberg [5]. NIPALS is shown in algorithm 1.

$A_{max}$ is the number of principal components one will attempt to extract. The convergence value, $tol$, is chosen as a small value based on the application and is typically smaller than $10^{-4}$. The start value of $t_s$ is not important as it will converge to the same value regardless of the initial values. However, a common practice is to choose the column with the biggest variance. In step 4 $E_{i-1}$ is projected onto $t_s$ and in step 5 $t_{new}$ is calculated from the projection of $E_{i-1}$ onto $p_i$. After convergence, in step 9, one subtracts the information modeled by the ith component from $E_{i-1}$. After the NIPALS, all values of $p$ and $t$ are gathered in the matrices $P$ and $T$.

**Algorithm 1** NIPALS PCA

```
$E_0 = \frac{X-1\bar{x}'}{\sigma}$ - //standardization of $X$
for $a = 1 : A_{max}$ do
    while $\epsilon > tol$ do
        $p_a = \frac{t'_s E_{a-1}}{t'_s t_s}$
        $p_a = ||p_a||$
        $t_{new} = \frac{E_{a-1} p_a}{p'_a p_a}$
        $\epsilon = \| t_s - t_{new} \|$
        $t_s = t_{new}$
    $t_a = t_{new}$
    $E_a = E_{a-1} - t_a p'_a$
```

## 2.2 SIMCA

SIMCA stands for *Soft Independent Modeling of Class Analogy* and is a classification method for testing samples on an existing PCA model. A paper comparing SIMCA with other methods such as PCA-LDA, neural network, and others [23] found that neural networks and other algorithms outperformed SIMCA in classification accuracy. However, SIMCA performed better then PCA-LDA. Both SIMCA and PCA-LDA are implemented in The Unscrambler X, but since the goal is not to compare classification methods, but instead investigate the use of classification in general, only SIMCA has been used in this thesis.

### 2.2.1 Projecting new samples

What SIMCA does is that it project new samples upon one or several existing models. This is done by projecting the samples onto the models' loadings. The new scores, $t_{new}$, is calculated by the new sample vector, $x_{new}$ and the loadings, $\mathbf{P}$, the following way:

$$t_{new} = x_{new} * \boldsymbol{P}(\boldsymbol{P}^T \boldsymbol{P})^{-1} \tag{2.4}$$

The new score and the sample residuals are calculated to determine if the sample is a part of the model. The residuals from the sample are calculated:

$$e_{new} = x_{new} - t_{new} \boldsymbol{P}^T \tag{2.5}$$

The residuals are described by CAMO Software as sample-to-model distance. Comparing one sample's residual variance, $S_i$, to the overall residual variation of the class, $S_0$, is done. If the new samples are within a limit, $S_{max}$, the sample is said to belong to the class. Another criterion that is tested is how the new score, $t_{new}$, fits the population of the other scores within the model. The calculations and criteria can be found in the method reference written by CAMO Software [7].

## 2.3 Data mining

Data mining is extracting useful patterns from large data and presenting them in an understandable way. This theory section is based on the book *Data mining: concepts and techniques* [14] which covers data mining from beginning to end. In the first chapter the book describes the data mining algorithm by the following steps:

- Data cleaning
- Data integration - Integrating data from multiple sources. Not relevant for this thesis.
- Data selection - Selecting which data to use. Not relevant for this thesis.
- Data transformation
- Data mining
- Pattern evaluation
- Knowledge presentation

### 2.3.1 Data cleaning

Data cleaning focuses solely on numbers. Is it information where it should be, and if so; how does this information stand in contrast to all other information? There are two main problems to be dealt with in the data cleaning phase according to the book: Noisy data and missing data. There are two different solutions to missing data: remove or bias. Removing the missing data problem can be done by removing the data all by itself or manually inserting the correct numbers. Manually inserting the correct amount of number can be extremely time-consuming and therefore not possible for large sets of data. The other way is to bias the data by inserting information based on a rule where there is missing data. This could be a global constant, the attribute mean or the most probable value. Regression or other statistical methods can be used to determine the most probable value. Noisy data is data created as a consequence of abnormal behavior or logging error. It could be a customer buying over 200% more than the average amount, and thus his/her behavior is not representative if one seeks to model the average customer. There are several methods to deal with this problem, however, in section 2.1.4 the influence of each sample in the PCA was mentioned. The influence plot conditions is an example of a method which captures and deals with this type of problem.

### 2.3.2 Data transformation

The book brings up five key-concepts in data transformation:

- Smoothing
- Aggregation
- Generalization
- Normalization
- Attribute construction

PCA and other transforms are examples of aggregation/generalization. The data is transformed by an algorithm which gives it new properties. Normalization and attribute construction revolves around transforming the range of the data by a universal rule. This is dependent on which algorithm one later wish to use and what criteria the data needs to satisfy for it to function correctly. Three examples are given in the book; min-max normalization, z-score normalization and decimal scaling. Min-max normalization remaps a value, v, within the min-max range of the data.

$$v_{mapped} = \frac{v - min_{data}}{max_{data} - min_{data}} * (newmax_{data} - newmin_{data}) + newmin_{data} \tag{2.6}$$

This method will be necessary for algorithms which needs the data to be within a specific range. Another normalization is called the z-score normalization. In data range A, the z-score $v_z$ is calculated by subtracting the original value, $v$, by subtracting the mean and dividing by the standard deviation.

$$v_z = \frac{v - \bar{A}}{\sigma_A} \tag{2.7}$$

The last normalization that is mentioned is decimal scaling which is diving each data sample by $10^j$.

$$v_{dec} = \frac{v}{10^j} \tag{2.8}$$

The z-score normalization results in the variables having zero mean and standard deviation equal to one. This property is useful in PCA when one has several different types of data in the variables, or if one wants to reduce the effect of variables operating at a higher range having more influence in the models. Min-max normalization and decimal scaling are not necessary to apply before PCA.

### 2.3.3 Pattern mining, evaluation, and presentation

After the data has been transformed into its final state the patterns revealed are put to the test. Are all patterns useful? This is often a subjective matter depending on which assumptions/goals one had before initializing the analysis. The book defines patterns as interesting ”if it is (1) *easily understood* by humans, (2) *valid* on new or test data with some degree of *,* (3) potentially *useful*, and (4) *novel*.” More objective measurements on pattern evaluations are mentioned as support and confidence. Support describes the association probability of two variables X and Y, $P(X \cup Y)$, while confidence the degree of certainty, $P(X|Y)$. Ones the patterns have been mined and evaluated they need to be presented in a way such that they can be acted upon. Results of the analysis are presented on what is called a *dashboard*, which is a presentation of results that can be used in business decisions.

Chapter 3

# Experiment

## 3.1 Introduction to experiment

This chapter discusses how the data was processed and formatted. The formatting was implemented in Python using a package named Pandas [2].

## 3.2 Hot pot soups - data processing

The data used is sales records from an online Chinese store. The online store, JD.com, is a business-to-customer market where the hot pot company markets their products. The name of the company is disclosed due to business purposes. Hotpot is a meal in China where one cooks different types of food in a bowl of simmering soup stock. The products are instant flavors to put in hot water, and a total of 40 different products are sold. The following columns structure the sales records:

- Product ID
- Quantity
- Account name
- Unit price
- Order price

- Settlement amount - This is the order price after discount (if there was any).
- Shipping province
- Purchase device - Computer/Phone/WeChat/QQ - WeChat and QQ are message-apps used in China.

Each row is one sale containing one product. If a customer were to buy several different products, these would show up as consecutive rows with the same account name and purchase time. The original dataset is not on a suitable format for performing PCA, and thus several different transformations have been applied.

### 3.2.1 Customer segmentation

With several of the models, the customers are not the subject of investigation, but other factors such as day and time. Still, in these models, several customers have been removed due to their behavior. As discussed in the theory section, there are several methods for handling noisy data. In discussions with Harald Martens, the conclusion when looking at some of the early models was that one customer group, in particular, should be removed. This group consists of the biggest spenders. Some accounts have made a purchase of 200 spice mixes in one day. That one day might affect the model in a strong way when one wishes to examine the increase in total interest, not one specific client's interest. All customers who had bought more than 30 of the same products in one purchase were removed. 30 was chosen by trial and error, and no specific rule was used. This resulted in the exclusion of 33 customers out of a total of 22536 customers.

### 3.2.2 Product segmentation

The products the company sells falls under these categories:

- Type
  - Hotpot Seasoning
  - Instant Spicy Steampot
  - Spicy Pot
  - Sichuan Food Seasoning
  - Hotpot Dip
- Continued/Discontinued

Associate Professor Yide Liu did these groupings, which gives a better understanding of the products. Different properties give possibilities for further investigation in the dataset. Today not all products are sold, as the ”continued/discontinued” products indicate. The records contain sales from 2016 and 2017, and all products which were sold in both years was kept. As a consequence, 28 products were kept for the analysis.

### 3.2.3 Data pre-prosessing

As discussed in the theory section, pre-processing is an essential step in data mining. In addition to the product and customer segmentation, the influence plot of the PCA analysis was used to detect and remove abnormalities. Within the software of The Unscrambler X, one can mark the outliers and recalculate without the marked samples.

### 3.2.4 Products and attributes

The products have been encoded by numbers ranging from 1-28 and their properties can be seen in table 3.1:

**Table 3.1:** Product attributes

| Product Nr. | Type (food) | Discontinued (2018) | Total sales |
|---|---|---|---|
| 1 | Hotpot Seasoning | Yes | 13889 |
| 2 | Instant Spicy Steampot | No | 565 |
| 3 | Instant Spicy Steampot | No | 3013 |
| 4 | Instant Spicy Steampot | No | 5057 |
| 5 | Spicy Pot | No | 2203 |
| 6 | Hotpot Seasoning | No | 2605 |
| 7 | Hotpot Seasoning | No | 5332 |
| 8 | Hotpot Dip | No | 6332 |
| 9 | Hotpot Seasoning | No | 822 |
| 10 | Hotpot Seasoning | No | 2237 |
| 11 | Hotpot Seasoning | No | 1592 |
| 12 | Spicy Pot | No | 2399 |
| 13 | Hotpot Dip | No | 3297 |
| 14 | Sichuan Food | No | 2253 |
| 15 | Sichuan Food | No | 963 |
| 16 | Instant Spicy Steampot | No | 555 |
| 17 | Hotpot Dip | No | 1289 |
| 18 | Hotpot Seasoning | No | 844 |
| 19 | Sichuan Food | No | 1883 |
| 20 | Instant Spicy Steampot | No | 1058 |
| 21 | Sichuan Food | Yes | 189 |
| 22 | Hotpot Dip | Yes | 793 |
| 23 | Sichuan Food | Yes | 358 |
| 24 | Hotpot Dip | Yes | 686 |
| 25 | Sichuan Food | Yes | 203 |
| 26 | Hotpot Seasoning | No | 352 |
| 27 | Hotpot Seasoning | Yes | 365 |
| 28 | Hotpot Seasoning | No | 2965 |

## 3.3 Suggested data formats

This section contains the reasoning and thought behind the different data structures that were made for PCA. Python functions for each transformation can be found in the appendix.

### 3.3.1 Daily product/province popularity

One approach is to transform the data such that one represents each column by a product or province and each row by a day in the year. The dataset then contains the number of sales for each product/province each day. This structure facilitates exploration of popularity both in products and dates. An example is that one could discover different behavior in the different provinces in China, e.g., that one province buys more than the others during a specific time of year. The transformation was done by looping through all the sales and placing the sales of each day on a new data matrix.

### 3.3.2 Daily averages of buy times

Another aspect to investigate is the purchase times. Creating a new data matrix where each row is represented by a 30 minutes time interval, which in total gives 48 intervals during the day, facilitates this. Each column is represented by either the different products or the provinces to which they were shipped. This structure facilitates the investigation of the purchase time-interval as scores and product/provinces as loadings. I.e., one might discover that some products are bought earlier than other, or differences between the different provinces.

### 3.3.3 Customer-product matrix

This is inspired by an article on product network analysis [20] which analyzed multiple baskets (shopping baskets in an online store) with the focus on network centrality. The study tested two types of data structures. One called customer-product matrix and another called product-product matrix. The goal was to investigate which products contained highest network centrality, meaning which products are sold the most together with other products. The study suggests that the costumer-product matrix can be used ”to provide appropriate products for customers when there is no costumer-related information” and that the product-product matrix is useful ”when we want to analyze the customer purchase preference in the long term.” The article focuses only on each products degree centrality, and not on how each product relate to one another. A similar data matrix can be analyzed with PCA. The costumer-product matrix used in this thesis will consist of all products that each customer have bought. It will not be just one single basket, but a combination of all basket each customer has purchased. The idea is to capture the different products commonly purchased together, and maybe achieve a grouping of costumers depending on their preference.

Chapter 4

# Analysis and results

## 4.1 Introduction to analysis

This chapter will contain different results and plots, but all results will be structured similarly. First, the explained variance is examined. Is it higher/lower than expected? Why? Outliers and residuals will be examined and interpreted early, mainly because these can give false contributions to the model. Then the scores and loadings will be discussed. What trends can one detect, and what are their causes.

## 4.2 Daily product sales

**Table 4.1:** Data structure example (values are fictional) - Daily product sales

| Day | Product 1 | Product 2 | ... | Product N |
|---|---|---|---|---|
| 01/01/2017 | 23 | 21 | | 5 |
| 02/01/2017 | 21 | 12 | | 2 |
| ... | ... | .. | | ... |
| 31/12/2016 | 10 | 12 | | 2 |

### 4.2.1 Standardization and pre-processing

The data, as shown in table 4.1, represents the total number of sales a given product had on a given day. A z-score transformation was used to transform the data. Standardizing the data prevents the popular products with many sales from dominating the model.

### 4.2.2 Explained Variance

In table 4.2 one can see the explained variance by the number of components. The maximum number of components was set to seven. One could theoretically have more, but as discussed in the theory section often only the first components will have a significant contribution. As seen in 4.2, one can tell that it is mainly the three first PCs that capture the variance in the data. From PC-3 and beyond one can see that the explained variance in the validation does not increase by a significant percent. This is an indication that PC-1,2 and three will be the ones of most importance. An explained variance of $\approx 30\%$ is not very high, but this is purchase patterns over the course of two years. A higher explained variance might not be realistic.

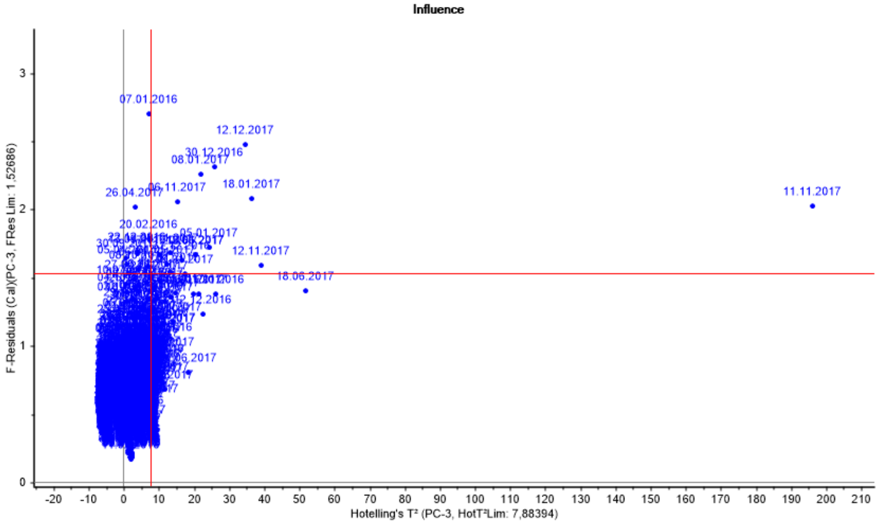
**Table 4.2:** Explained variance - Daily Sales

| | PC-0 | PC-1 | PC-2 | PC-3 | PC-4 | PC-5 | PC-6 | PC-7 |
|---|---|---|---|---|---|---|---|---|
| Calibration | 0 | 24,93 | 34,59 | 39,43 | 43,81 | 47,99 | 51,91 | 55,58 |
| Validation | 0 | 20,84 | 27,91 | 29,25 | 30,14 | 30,83 | 32,33 | 34,21 |

### 4.2.3 Influence plot

In figure 4.1 one can see that most of the scores, which represents days, are described well by the model. However, there are some outliers. The biggest outlier is the 11.11.2017. This day is known as Singles' Day in China, which is a day dedicated to celebrating being single. Singles' Day in 2017 beat all previous records with shoppers spending more than $25bn [13]. In other words, it is an outlier, but it is also a key factor in describing this day purchase behavior in China. It should not be removed. Several other days stand out. E.g., 12.11.2017, the day after 11.11.2017, and 12.12.2016/2017 which is Cyber Monday. The sales records of the outlier dates were examined in search of abnormalities such as logging errors, but nothing special was unveiled.

**Figure 4.1** Influence plot of the daily sales PCA model. Three components has been included in the model, as the PC-1,2 and three showed the biggest contributions.



## 4.2.4 Scores and loadings

Looking at the correlation loadings in figure 4.2, it would seem that the first principal component is related to popularity. In fact, products 1,8,7,4,13 is respectively the five most selling products during the 2016/2017, as seen in table 3.1. All which is placed high on PC-1. If one looks at the scores plot in figure 4.3 the highest value on PC-1 is Singles' day (11.11.2017). This strengthens the theory of PC-1s' association with popularity. Solely looking at the dates of the scores can be overwhelming. There is just too much to analyze each day in two years. However, these dates share several common factors and can be grouped by their properties.

**Figure 4.2** Correlation loadings - Daily Sales. PC-1,2.

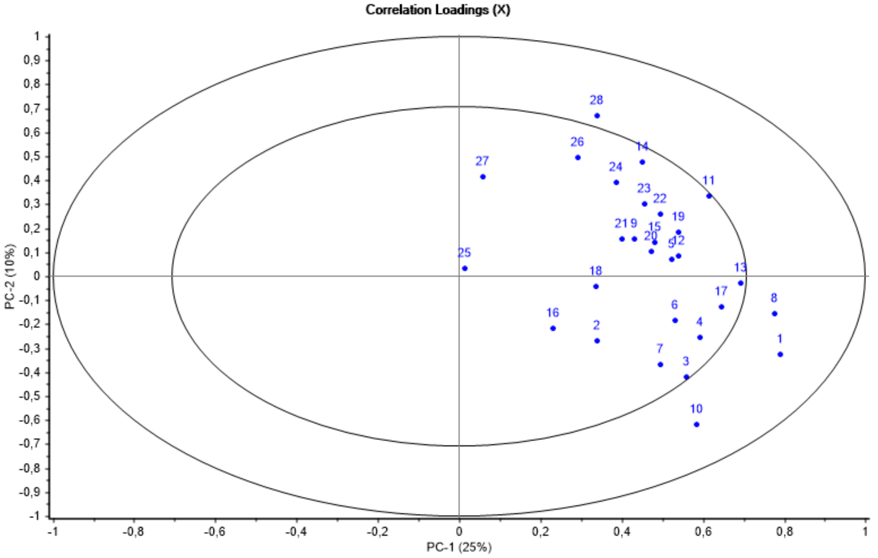
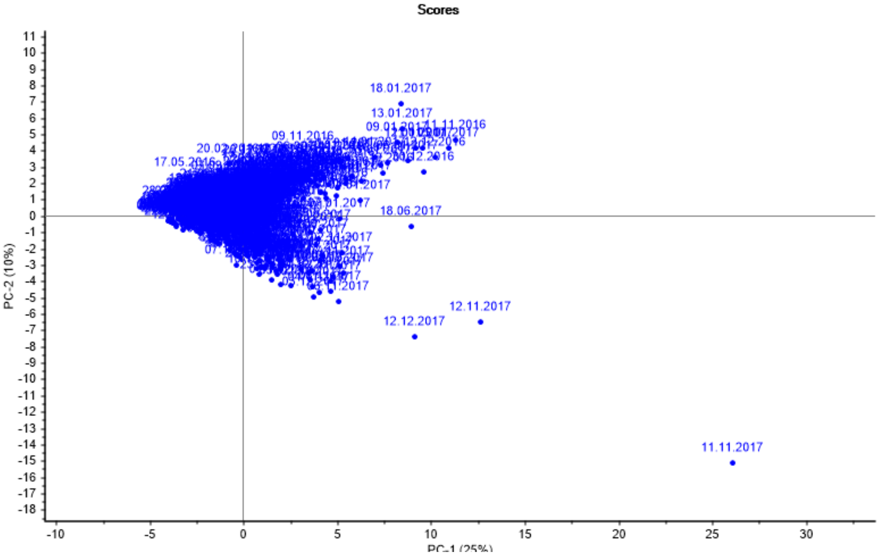


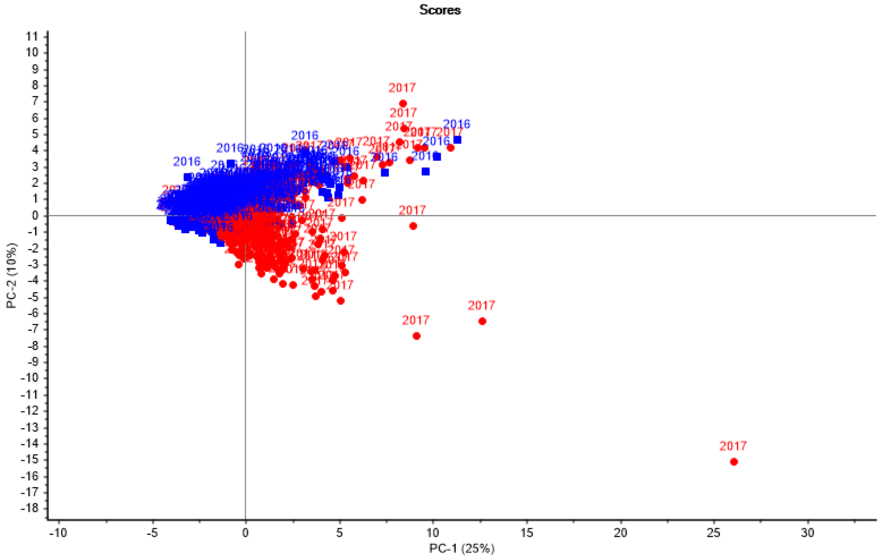
**Figure 4.3** Scores - Daily Sales. PC-1,2



### 4.2.5 Sample groupings - Year

In figure 4.4 the scores have been marked based on the year. There is a clear distinction between 2016 and 2017 in the fact that 2016 has very few points in the bottom right quadrant. The purchase pattern of the customers has changed as they are now buying more of the products located in the bottom right corner of the loadings than they did in 2016.

**Figure 4.4** Scores - Daily sales. PC-1,2. Grouped by year. Red scores are made by the samples from 2017, and the blue from 2016.



This distinction can be seen together with the products which we know has been discontinued. These are marked in the correlation loading plot seen in figure 4.5. All discontinued products lie with the upper right quadrant and might be the consequence of the pattern visible in figure 4.4. However, when recalculating the PCA without these products, one obtains a similar pattern in the scores as seen in figure 4.4. This pattern can be seen in figure 4.6.

**Figure 4.5** Loadings plot - Daily sales. PC-1,2. Marked products are the products which are now discontinued.

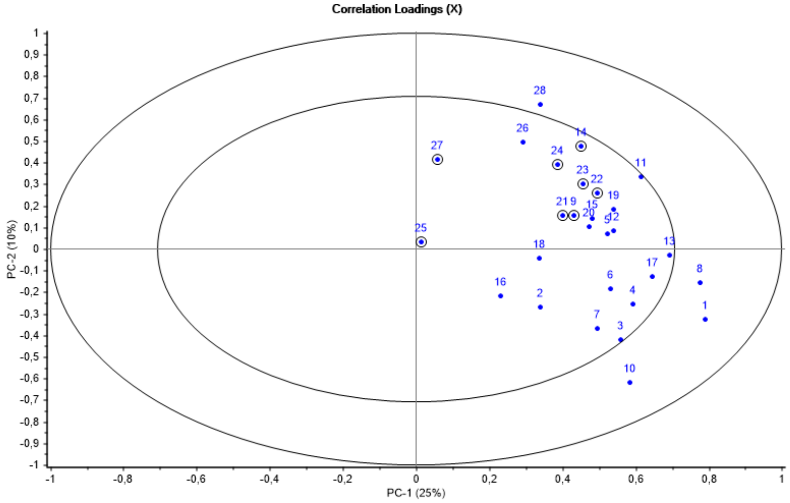
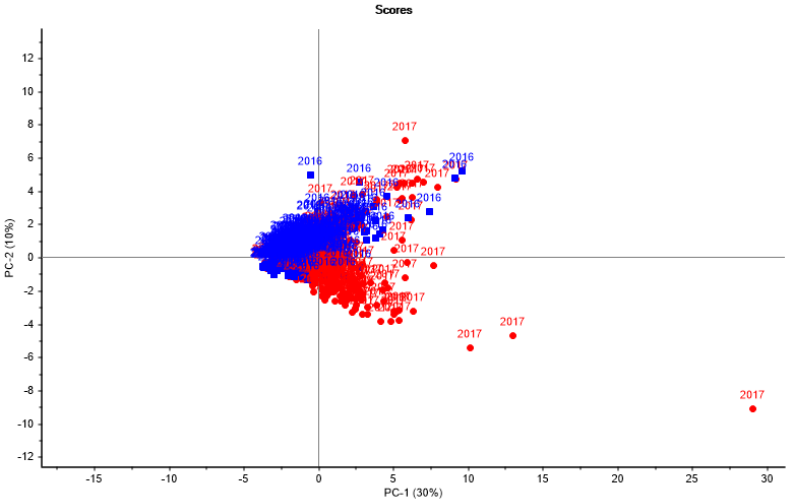


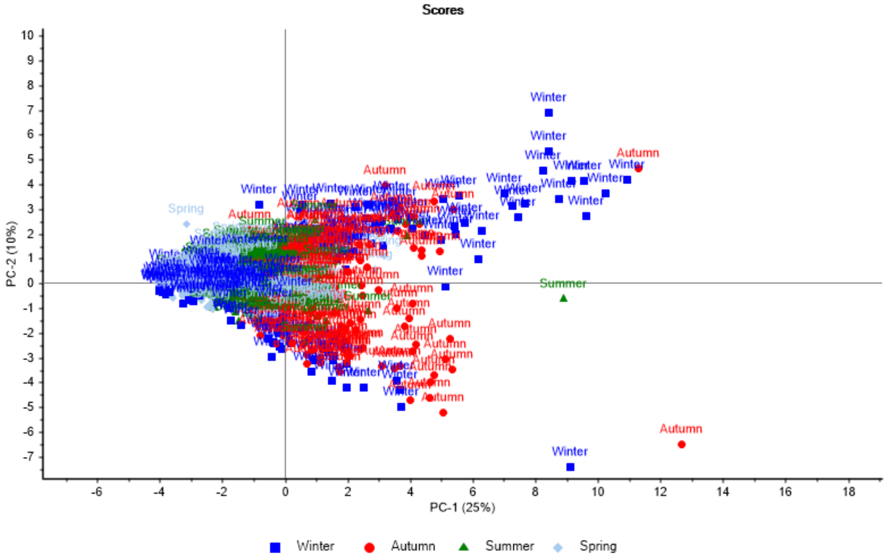
**Figure 4.6** Scores plot - marked by year, without discontinued products. PC-1,2. The same pattern as in 4.4 can be seen, with a new direction taking place in 2017.



### 4.2.6 Sample groupings - Seasons

Another effect to investigate under sample grouping is the seasons. This grouping of the scores can be seen in figure 4.7. The seasons in China is winter (December - February), spring (March-May), summer (June - August) and autumn (September - Oktober). From the scores in figure 4.7 it is clear that winter and autumn are the most popular seasons as the days in these seasons lie the rightmost on PC-1. One day, 18.06.2017, did exceptionally well for being during summer. It turned out that on this particular day (the green summer point located approximately at nine on PC-1 in figure 4.7) the company did a promotion. Some days in the winter did exceptionally bad and had the highest negative values on PC-1. These were the days leading up to the Chinese new year. The dates with the highest negative PC-1 score was: 06.02.2016, 09.02.2016, 27.01.2017, 10.02.2016, 03.02.2016, 02.02.2016, 30.01.2017 and 26.01.2017. The dates of new year were for 08.02.2016 and 28.01.2017 [4].

**Figure 4.7** Scores - Daily sales. PC-1,2. Grouped by season.



In figure 4.8 one can see the scores plot of PC-2 and PC-3. PC-2 has already shown some linkage to the difference between the products that sold well in 2016 and those that sold well in 2017. In PC-3 we see a difference between the seasons where autumn/winter tends to score high on PC-3 and summer/spring scores lower. The average PC-3 score value for the winter, spring, summer and autumn in respectively 0.51, -0.11, -0.56 and 0.166. In figure 4.9 one can see the correlation loadings for PC-2,3. The marked products are under the category ”hotpot seasoning.” This was the only visible pattern in the correlation loadings, as it seems that these are the most popular during autumn/winter. The other categories are responsible for the negative direction on PC-3, and thus might be more popular during the summer/spring.

**Figure 4.8** Scores - Daily sales. Grouped by season, PC-2,3

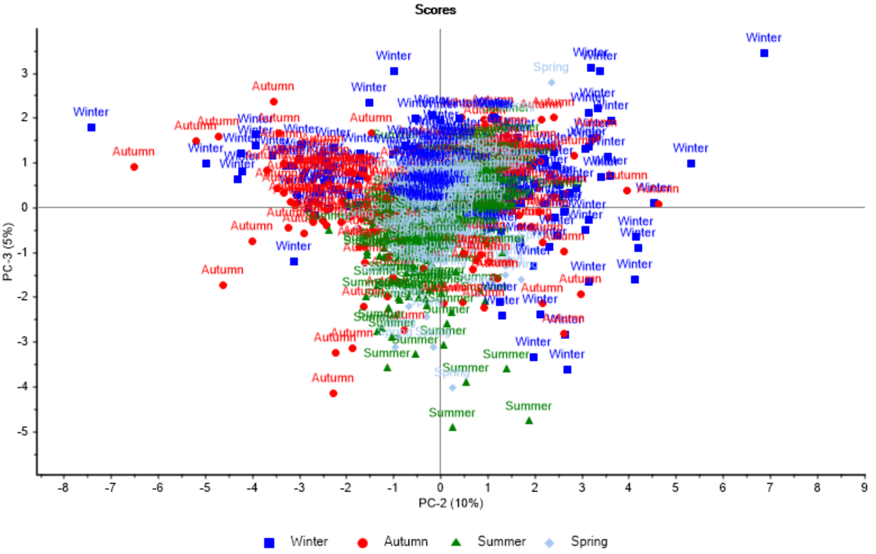
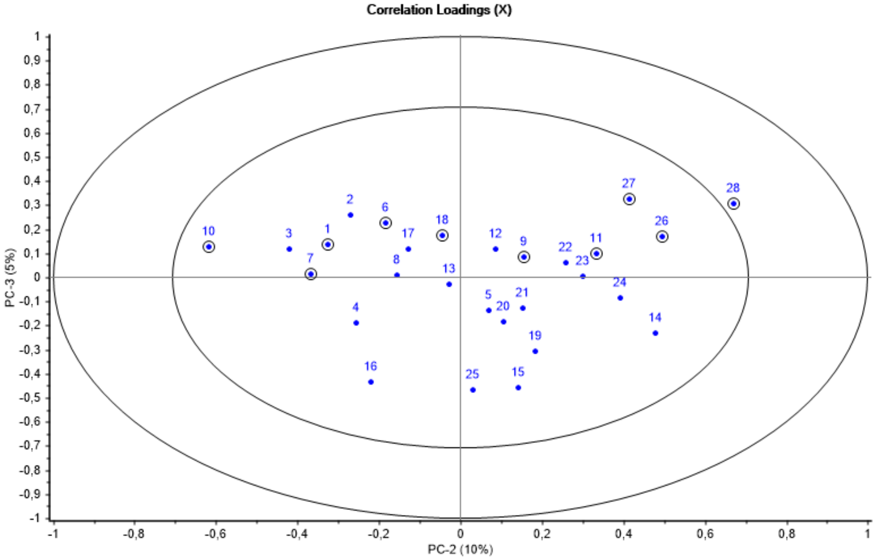


**Figure 4.9** Loadings - Daily sales. PC-2,3. Marked products are under the category ”hotpot seasoning”



## 4.3 Daily province sales

The daily province sales was formatted in a similar way as the daily products and its structure is shown in table 4.3:

**Table 4.3:** Data structure example of daily province sales.

| Day | Province 1 | Province 2 | ... | Province N |
|---|---|---|---|---|
| 01/01/2017 | 23 | 21 | | 5 |
| 02/01/2017 | 21 | 12 | | 2 |
| ... | ... | .. | | ... |
| 31/12/2016 | 10 | 12 | | 2 |

### 4.3.1 Data and pre-processing

As with the daily product sales, a z-score transformation was used to eliminate the most-buying provinces consuming too much of the model. A summary of the data can be found in table 4.4. It contains the different provinces that had recorded orders from the store. ”Coast/Inland” and ”Location” has been manually set by looking at the map. ”Habitants” comes from each provinces Wikipedia page.

**Table 4.4:** Province sales - collected data.

| Column position | Name | Coast/Inland | Location | Habitats (million) | Total shipped products | (Total shipped products)/Habitats (million) |
|---|---|---|---|---|---|---|
| 1 | Shanghai | Coast | East Center | 23 | 3730 | 162.17 |
| 2 | Shaanxi | Inland | East Center | 37 | 4853 | 131.16 |
| 3 | Henan | Inland | East Center | 94 | 2595 | 27.61 |
| 4 | Sichuan | Inland | South Center | 80 | 8606 | 107.57 |
| 5 | Guangxi | Coast | South South | 46 | 685 | 14.89 |
| 6 | Shadong | Coast | East East | 91 | 2979 | 32.74 |
| 7 | Hubei | Inland | East Center | 57 | 2267 | 39.77 |
| 8 | Beijing | Inland | East East | 19 | 5169 | 272.05 |
| 9 | Fujian | Coast | East (South) | 36 | 1424 | 39.56 |
| 10 | Chongqing | Inland | Center Center | 28 | 1070 | 38.21 |
| 11 | Jiangxi | Inland | South East | 44 | 583 | 13.25 |
| 12 | Jilin | Coast | North East | 27 | 889 | 32.93 |
| 13 | Zhejiang | Coast | East East | 54 | 2441 | 45.20 |
| 14 | Jiangsu | Coast | East East | 78 | 3247 | 41.63 |
| 15 | Guangdong | Coast | South South | 104 | 6657 | 64.01 |
| 16 | Hebei | Coast | North East | 71 | 1905 | 26.83 |
| 17 | Gansu | Inland | North Center | 25 | 1756 | 70.24 |
| 18 | Liaoning | Coast | North East | 43 | 1539 | 35.79 |
| 19 | Heilongjiang | Inland | North (North East) | 38 | 933 | 24.55 |
| 20 | Hunan | Inland | South Center | 65 | 877 | 13.49 |
| 21 | Hainan | Coast | South South | 8 | 412 | 51.50 |
| 22 | Tianjin | Coast | East North | 12 | 947 | 78.92 |
| 23 | Yunnan | Inland | South South | 45 | 1270 | 28.22 |
| 24 | Shanxi | Inland | North East | 35 | 1035 | 29.57 |
| 25 | Anhui | Inland | East East | 59 | 950 | 16.10 |
| 26 | Ningxia | Inland | North Center | 6 | 495 | 82.50 |
| 27 | Guizhou | Inland | South South | 34 | 737 | 21.68 |
| 28 | Xinjiang | Inland | North West | 21 | 2773 | 132.05 |
| 29 | Indre Mongolia | Inland | North North | 24 | 776 | 32.33 |
| 30 | Tibet | Inland | South West | 3 | 251 | 83.67 |
| 31 | Qinghai | Inland | West | 5 | 237 | 47.40 |

### 4.3.2 Influence plot

The influence plot seen in figure 4.10 have many of the similar outliers dates as the daily product plot 4.1. One clear difference though is the marked sample, 30.03.2017. In the raw data, one can see that this is because on this particular day one person made a large purchase to the province which buys the least hot pot soups. As a consequence, the contribution of this day to the model the day should be removed from the analysis.

Figure 4.11 shows the influence plot of the PCA recalculated without the 30.03.2017 sample. The similar samples as in figure 4.1 are outliers, but as discussed these outliers have a valid reason and are not defined by just one customer.

**Figure 4.10** Influence plot - Daily province sale. PC-1

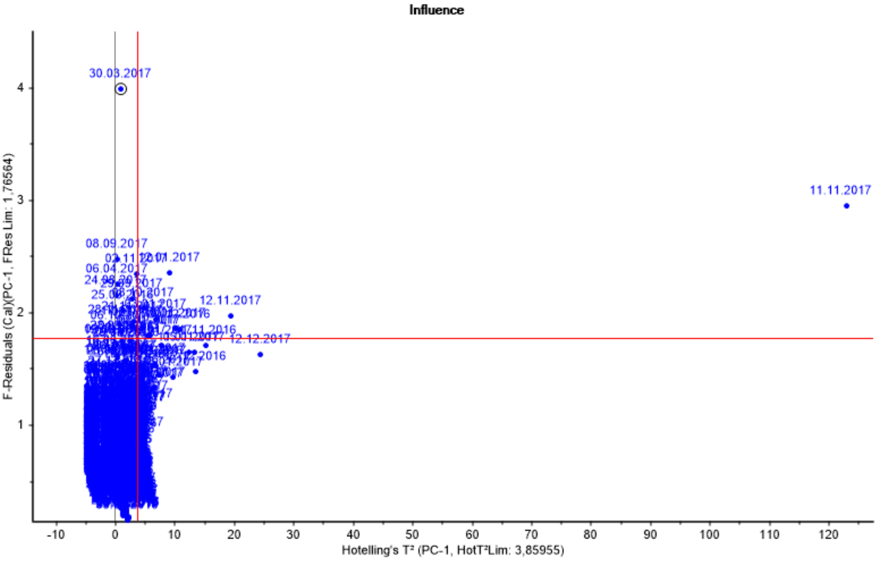
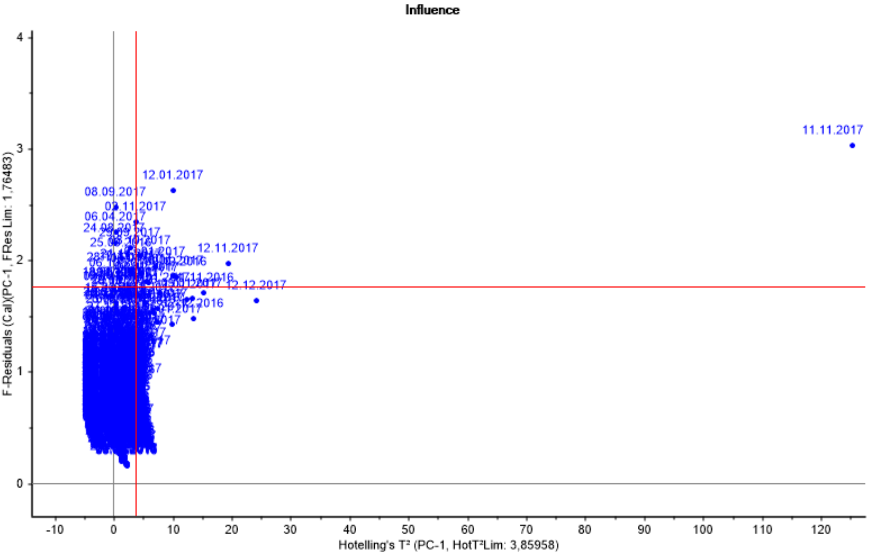


**Figure 4.11** Influence plot - recalculated without 30.03.2017. PC-1.



### 4.3.3 Explained variance

The variance captured by the model can be seen in table 4.5. Not much of the variance is captured within the first components, and one thing to take note is that the explained variance in the validation decreases after the first component. This is an indication that the PCA does not adequately capture structure after the first component. Compared to the previous analysis, the daily product sales, the PCA captured less explained variance in the daily province sales. There is less structure in which province the purchase was made, with the maximum explained variance being 15,30%.

**Table 4.5:** Explained variance - Province sales. In the validation, one can see how the explained variance decreases after PC-1, indicating that adding additional components does not improve the model.

| | PC-0 | PC-1 | PC-2 | PC-3 | PC-4 | PC-5 | PC-6 | PC-7 |
|---|---|---|---|---|---|---|---|---|
| Calibration | 0 | 19,08 | 23,51 | 27,59 | 31,59 | 35,46 | 39,18 | 42,69 |
| Validation | 0 | 15,30 | 14,92 | 14,31 | 14,34 | 14,92 | 15,19 | 14,82 |

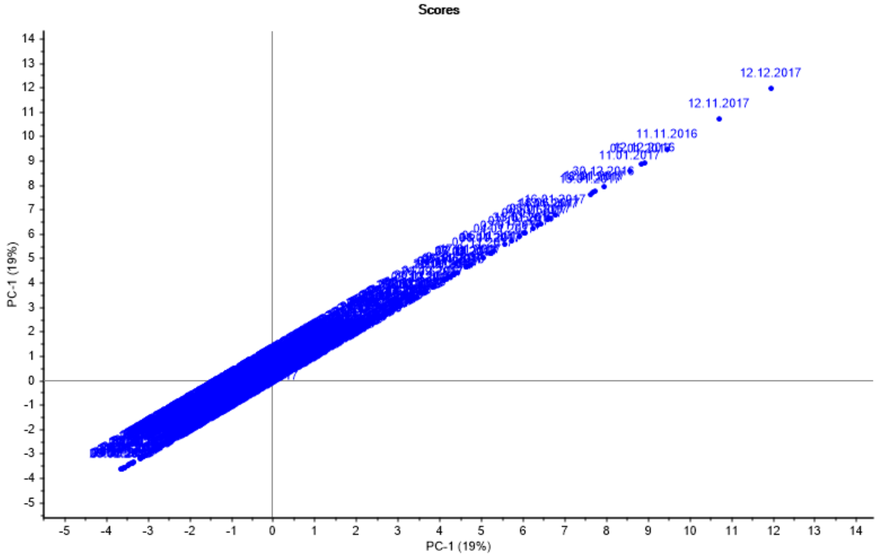
### 4.3.4 Scores and loadings

Given that there only lies information in PC-1, a table was used for investigating rather than the loadings plot. In table 4.6 one can see table 4.4 sorted by the loading values in descending order. One property that can be seen is that in general, the most selling products have the highest loading values, but this is not a definite trend. Shaanxi had the highest loading value but is not the province which buys the most. Other explanations cloud possibly exist for PC-1. However, these were not visible based on available information. The score plot for PC-1 can be seen in 4.12. As in 4.4 one can see that the same samples, the ones representing a promotion day, has a large value. No patterns concerning coast/inland or location were found.

**Table 4.6:** Province names and attributes. Sorted by which province had the highest value on PC-1.

| Column position | Name | Coast/Inland | Location | Habitats (million) | Total shipped products | (Total shipped products)/Habitats | PC-1 Loading |
|---|---|---|---|---|---|---|---|
| 2 | Shaanxi | Inland | East Center | 37 | 4853 | 131.162 | 0.2935 |
| 4 | Sichuan | Inland | South Center | 80 | 8606 | 107.575 | 0.2692 |
| 7 | Hubei | Inland | East Center | 57 | 2267 | 39.7719 | 0.2457 |
| 8 | Beijing | Inland | East East | 19 | 5169 | 272.053 | 0.2453 |
| 15 | Guangdong | Coast | South South | 104 | 6657 | 64.0096 | 0.2420 |
| 28 | Xinjiang | Inland | North West | 21 | 2773 | 132.047 | 0.2327 |
| 14 | Jiangsu | Coast | East East | 78 | 3247 | 41.6282 | 0.2267 |
| 3 | Henan | Inland | East Center | 94 | 2595 | 27.6063 | 0.2240 |
| 6 | Shadong | Coast | East East | 91 | 2979 | 32.7362 | 0.2231 |
| 17 | Gansu | Inland | North Center | 25 | 1756 | 70.2400 | 0.2185 |
| 18 | Liaoning | Coast | North East | 43 | 1539 | 35.7906 | 0.2076 |
| 16 | Hebei | Coast | North East | 71 | 1905 | 26.8309 | 0.2006 |
| 25 | Anhui | Inland | East East | 59 | 950 | 16.1016 | 0.1891 |
| 1 | Shanghai | Coast | East Center | 23 | 3730 | 162.173 | 0.1860 |
| 13 | Zhejiang | Coast | East East | 54 | 2441 | 45.2037 | 0.1809 |
| 24 | Shanxi | Inland | North East | 35 | 1035 | 29.5714 | 0.1635 |
| 23 | Yunnan | Inland | South South | 45 | 1270 | 28.2222 | 0.1588 |
| 29 | Indre Mongolia | Inland | North North | 24 | 776 | 32.3333 | 0.1556 |
| 11 | Jiangxi | Inland | South East | 44 | 583 | 13.25 | 0.1454 |
| 9 | Fujian | Coast | East | 36 | 1424 | 39.5556 | 0.1375 |
| 5 | Guangxi | Coast | South South | 46 | 685 | 14.8913 | 0.1359 |
| 20 | Hunan | Inland | South Center | 65 | 877 | 13.4923 | 0.1176 |
| 27 | Guizhou | Inland | South South | 34 | 737 | 21.6764 | 0.1173 |
| 22 | Tianjin | Coast | East North | 12 | 947 | 78.9167 | 0.1057 |
| 12 | Jilin | Coast | North East | 27 | 889 | 32.9259 | 0.1034 |
| 10 | Chongqing | Inland | Center Center | 28 | 1070 | 38.2143 | 0.0979 |
| 19 | Heilongjiang | Inland | North (North East) | 38 | 933 | 24.5526 | 0.0860 |
| 26 | Ningxia | Inland | North Center | 6 | 495 | 82.5000 | 0.0848 |
| 30 | Tibet | Inland | South West | 3 | 251 | 83.6667 | 0.0841 |
| 31 | Qinghai | Inland | West | 5 | 237 | 47.4000 | 0.0836 |
| 21 | Hainan | Coast | South South | 8 | 412 | 51.5000 | 0.0480 |

**Figure 4.12** Province daily sales - Scores



## 4.4 Product purchase times

In this section, the principal components of the total purchases within different time intervals will be investigated. The data was structured as in table 4.7. All data was standardized with a z-score standardization in to eliminate the dominant effect of popular products in the model.

**Table 4.7:** Data structure example of product purchase times.

| Time interval | Product 1 | Product 2 | ... | Product N |
|---|---|---|---|---|
| 00:00-00:30 | 23 | 21 | | 5 |
| 00:30-01:00 | 21 | 12 | | 2 |
| ... | ... | .. | | ... |
| 23:30-00:00 | 10 | 12 | | 2 |

### 4.4.1 Explained variance

The explained variance captured by the PCA model can be seen in table 4.8. The first principal component alone captures almost the total variation in the dataset. In the validation, nearly 79% of the variance was explained by the first component. The following components add approximately 1% explained variance for each component. This indicates that it is mostly the first component which contains an analytic value, and the others may just be modeling rare occurrences or noise.

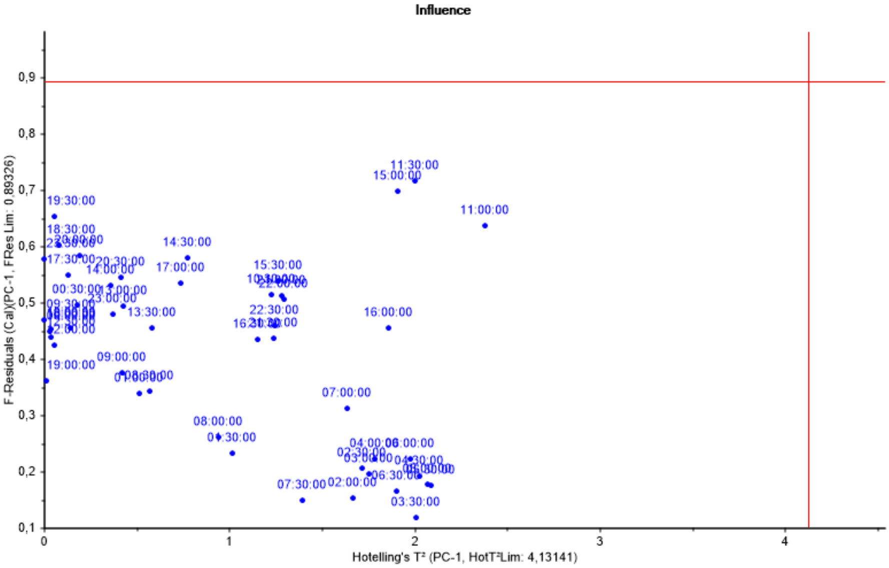
**Table 4.8:** Explained Variance - Product Purchase Times

| | PC-0 | PC-1 | PC-2 | PC-3 | PC-4 | PC-5 | PC-6 | PC-7 |
|---|---|---|---|---|---|---|---|---|
| Calibration | 0 | 79,84 | 83,17 | 85,71 | 87,82 | 89,80 | 91,37 | 92,52 |
| Validation | 0 | 78,88 | 79,85 | 80,51 | 81,01 | 82,98 | 83,98 | 84,06 |

### 4.4.2 Influence plot

In figure 4.13 one can see the influence plot of the first principal component. There are no outliers.

**Figure 4.13** Influence plot - Product purchase times. PC-1



### 4.4.3 Scores and loadings

In figure 4.14 one can see that there is a distinction between the positive and negative value on PC-1. The blue line is connecting the consecutive samples. It can be followed from 00:00 to 23:30. One can see that from 00:00 to 09:30 there are mainly negative score values, while from 09:30 to 23:30 the samples have positive PC-1 scores. When one sees the scores in context with table 4.9 it is clear that a higher score on PC-1 indicates that the purchase time is more popular. The time intervals with the most sales (11:00, 11:30, 15:00 and 16:00) also have the highest scores on PC-1. On the loadings plot, seen in figure 4.15, the marked variables are the seven products that sold the least in 2016/17. These are the products that have the highest/lowest values on PC-2 and indicates that PC-2 seems to model the products which have the least data. The fact that PC-2 only contributes with 1% explained variance supports this argument.

**Figure 4.14** Scores - Product Purchase Times. PC-1,2. The line starts at 00:00 and ends at 23:30. Each time indicates a span of 30 minutes.

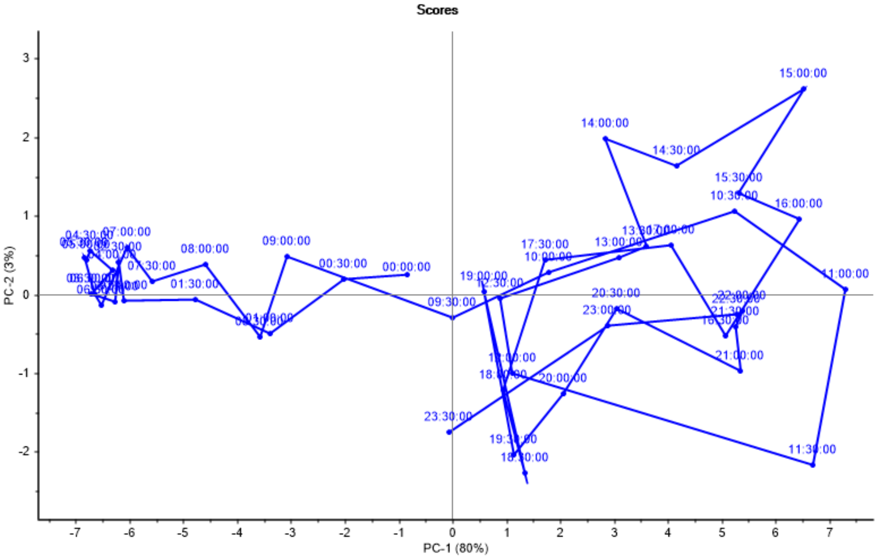


**Figure 4.15** Loadings - Product Purchase Times. PC-1,2. Seven least-selling products are marked.

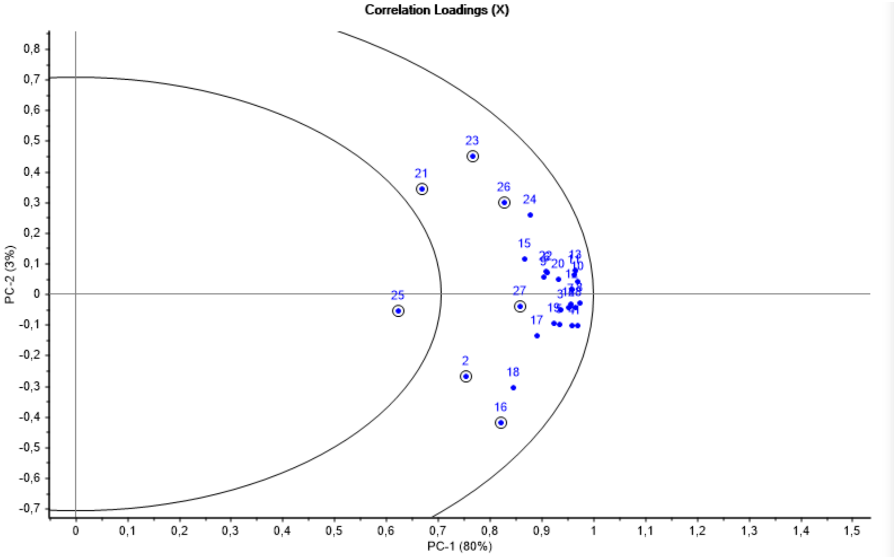


**Table 4.9:** Total number of sales in different time intervals. Time value indicates start of 30 minute interval.

| 30 minute interval, start time | Total sales 2016/2017 | 30 minute interval, start time | Total sales 2016/2017 |
|---|---|---|---|
| 00:00:00 | 1207 | 12:00:00 | 1692 |
| 00:30:00 | 899 | 12:30:00 | 1513 |
| 01:00:00 | 673 | 13:00:00 | 1914 |
| 01:30:00 | 444 | 13:30:00 | 1919 |
| 02:00:00 | 287 | 14:00:00 | 1785 |
| 02:30:00 | 261 | 14:30:00 | 2029 |
| 03:00:00 | 212 | 15:00:00 | 2238 |
| 03:30:00 | 163 | 15:30:00 | 2229 |
| 04:00:00 | 187 | 16:00:00 | 2460 |
| 04:30:00 | 141 | 16:30:00 | 2344 |
| 05:00:00 | 121 | 17:00:00 | 2060 |
| 05:30:00 | 80 | 17:30:00 | 1702 |
| 06:00:00 | 161 | 18:00:00 | 1583 |
| 06:30:00 | 185 | 18:30:00 | 1568 |
| 07:00:00 | 238 | 19:00:00 | 1579 |
| 07:30:00 | 360 | 19:30:00 | 1536 |
| 08:00:00 | 494 | 20:00:00 | 1722 |
| 08:30:00 | 665 | 20:30:00 | 1964 |
| 09:00:00 | 776 | 21:00:00 | 2249 |
| 09:30:00 | 1301 | 21:30:00 | 2323 |
| 10:00:00 | 1660 | 22:00:00 | 2455 |
| 10:30:00 | 2207 | 22:30:00 | 2287 |
| 11:00:00 | 2491 | 23:00:00 | 1877 |
| 11:30:00 | 2523 | 23:30:00 | 1335 |

## 4.5 Province purchase times

The province purchase times were organized as the product purchase times. An example can be seen in table 4.10. A z-score transformation was used to standardize the data.

**Table 4.10:** Data structure example (values are fictional)

| Time interval | Province 1 | Province 2 | ... | Province N |
|---|---|---|---|---|
| 00:00-00:30 | 23 | 21 | | 5 |
| 00:30-01:00 | 21 | 12 | | 2 |
| ... | ... | .. | | ... |
| 23:30-00:00 | 10 | 12 | | 2 |

### 4.5.1 Explained variance

The explained variance, as seen in table 4.11, is similar to one of the product purchase times 4.8. The first component captures the most of the data while the following components add little to no additional increase in explained variance. These results are similar to when the products purchase time was analyzed in the previous section, however, the explained variance in lower. It was the same for the daily province sales vs. daily product sales. There is more common variation between the products than the provinces.

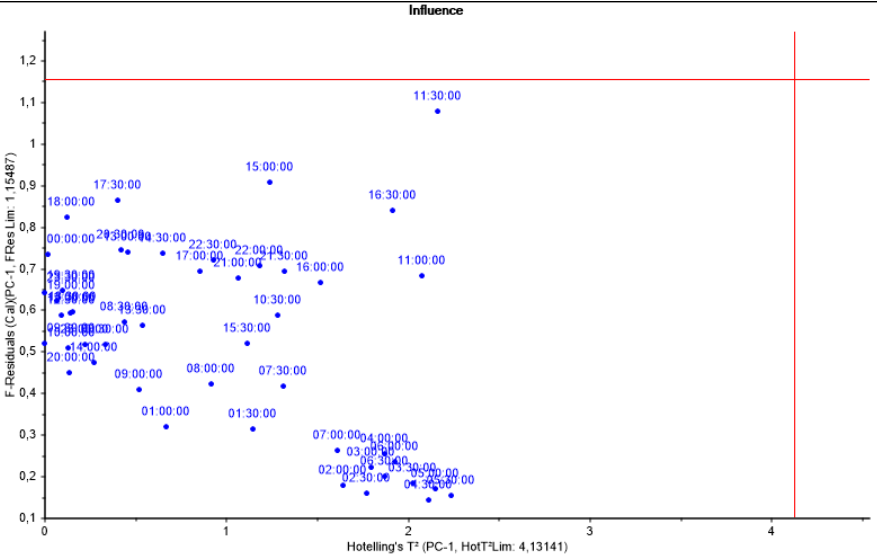
**Table 4.11:** Explained Variance - Province Purchase Times

| | PC-0 | PC-1 | PC-2 | PC-3 | PC-4 | PC-5 | PC-6 | PC-7 |
|---|---|---|---|---|---|---|---|---|
| Calibration | 0 | 66,31 | 71,02 | 74,88 | 78,31 | 81,01 | 83,54 | 85,85 |
| Validation | 0 | 62,97 | 63,99 | 63,94 | 65,47 | 64,85 | 65,26 | 67,11 |

### 4.5.2 Influence plot

There were no outliers, as seen in figure 4.16.

**Figure 4.16** Influence Plot - Province Purchase Times. No outliers were found. PC-1.



## 4.5.3 Scores and loadings

Looking at figure 4.17, one can see similarities with the score plot for the product purchase times 4.14. Many of the same time intervals, such as 11:00 and 11:30, have the highest values on PC-1. Also in the loading plot 4.18 one can witness the same phenomena. The three provinces making PC-2 is 31,30 and 21 which also is the provinces that buy the least, as seen in table 4.4. Again, this is an indication that the principal components following PC-1 do not capture any important structure.

**Figure 4.17** Scores - Province Purchase Times. Lines between all samples, starting at 00:00.

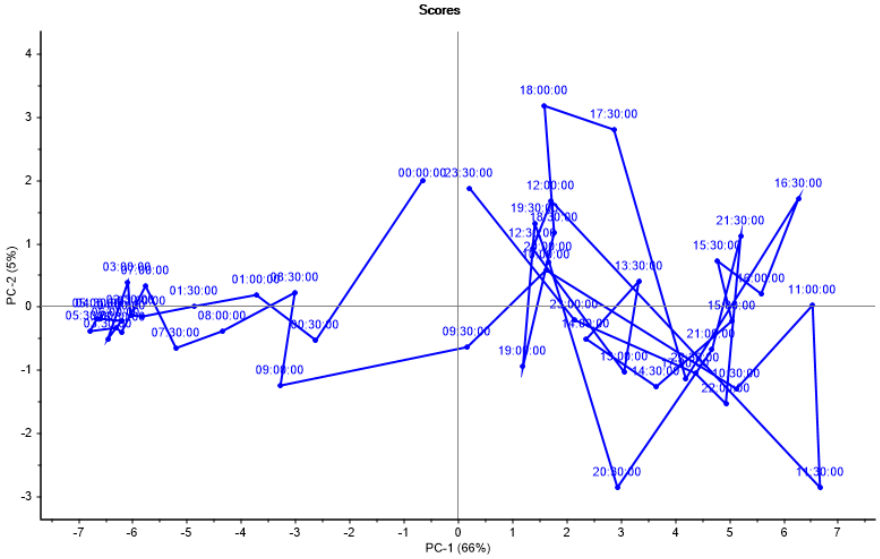
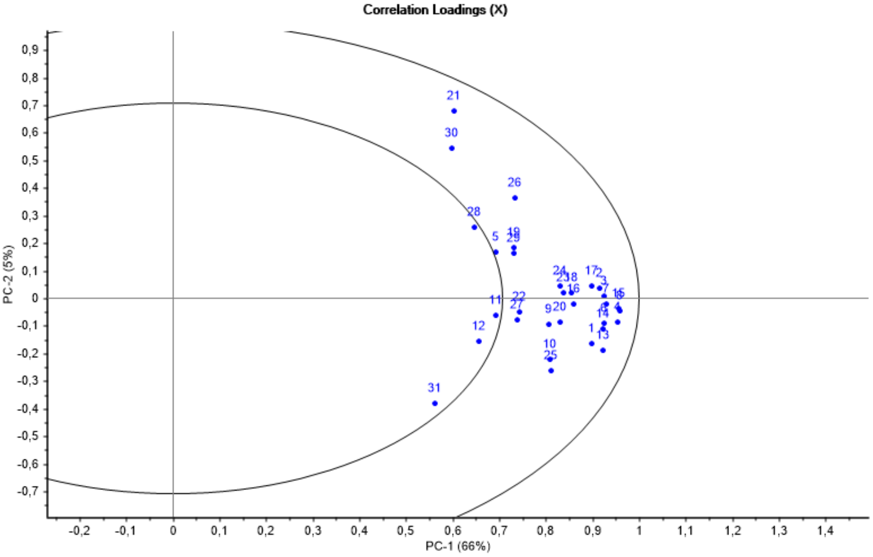


**Figure 4.18** Loadings - Province Purchase Times



## 4.6 Customer-product matrix

In table 4.12, one can see an example of the customer-product matrix structure.

**Table 4.12:** Costumer-product matrix - Example

| Costumer | Product 1 | ... | Product N |
|---|---|---|---|
| Costumer 1 | 23 | | 5 |
| ... | ... | | ... |
| Costumer N | 10 | | 2 |

Out of the 22147 customers that have bought products in 2016/2017 there were mostly single time buyers. A table showing how many costumers made how many purchases can be seen in table 4.13.

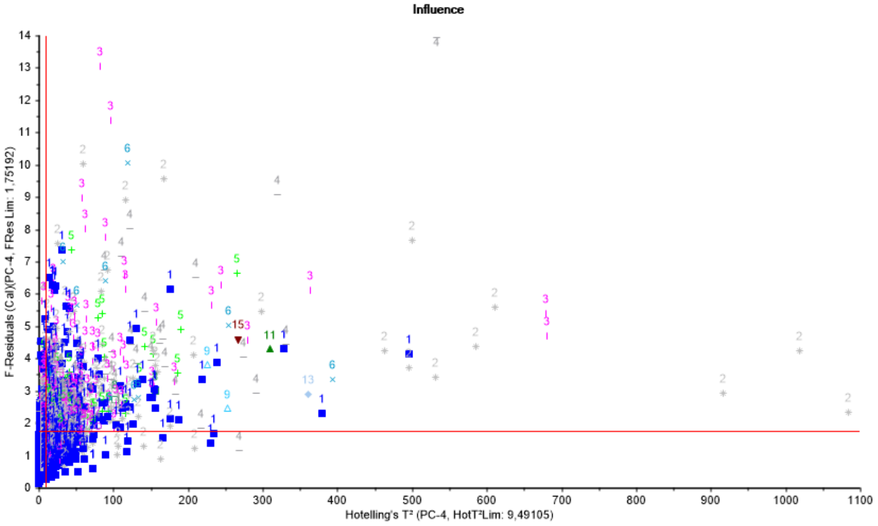
**Table 4.13:** Number of purchases

| Number of purchases | Accounts |
|---|---|
| 1 | 19221 |
| 2 | 2170 |
| 3 | 470 |
| 4 | 159 |
| 5 | 74 |
| 6 | 28 |
| 7 | 12 |
| 9 | 3 |
| 8 | 3 |
| 11 | 2 |
| 10 | 2 |
| 15 | 1 |
| 13 | 1 |

### 4.6.1 Influence plot

In the influence plot (figure 4.19) the samples (customers) are marked by the number of purchases they have made. There are several outliers, and among the outliers, it is clear that many of them consist of customers who have made several purchases at the site. Instead of removing all the outliers outside the red lines, the customers were segmented based on how many purchases they made. The groupings were: 1 purchase, 2-4 purchases, and 5 or more purchases.

**Figure 4.19** Influence plot - Customer-Product Matrix



## 4.6.2 Explained variance

From table 4.14 one can see that the explained variance in the validation does not capture more than approximately 2,2% of the variance for the customers who made one purchase. For the customers who made 2-4 purchases (table 4.15) 4,60% is the highest explained variance in the validation, taking into account three principal components. Customers who had 5 or more purchases did not get modeled at all by the PCA, as seen in table 4.16. The analysis stopped here due to the amount of variance captured by the model.

**Table 4.14:** Explained Variance - Customer-Product Matrix. One purchase

| | PC-0 | PC-1 | PC-2 | PC-3 | PC-4 | PC-5 | PC-6 | PC-7 |
|---|---|---|---|---|---|---|---|---|
| Calibration | 0 | 6,89 | 12,33 | 17,58 | 22,78 | 27,89 | 32,99 | 38,04 |
| Validation | 0 | 2,19 | 2,08 | 1,64 | 1,13 | 0,05 | -1,41 | -2,56 |

**Table 4.15:** Explained Variance - Customer-Product Matrix. 2-4 purchases.

| | PC-0 | PC-1 | PC-2 | PC-3 | PC-4 | PC-5 | PC-6 | PC-7 |
|---|---|---|---|---|---|---|---|---|
| Calibration | 0 | 6,41 | 11,83 | 16,95 | 21,47 | 25,72 | 29,83 | 33,88 |
| Validation | 0 | 1,97 | 2,84 | 4,60 | 4,46 | 3,61 | 2,73 | 2,33 |

**Table 4.16:** Explained Variance - Customer-Product Matrix. 5 or more purchases.

| | PC-0 | PC-1 | PC-2 | PC-3 | PC-4 | PC-5 | PC-6 | PC-7 |
|---|---|---|---|---|---|---|---|---|
| Calibration | 0 | 8,06 | 15,60 | 22,18 | 28,18 | 33,75 | 39,09 | 43,95 |
| Validation | 0 | -0,66 | 0,30 | 0,29 | -1,64 | -3,63 | -4,70 | -7,23 |

## 4.7 Projecting new samples with SIMCA

### 4.7.1 Daily product sales

In the previous sections, several models have been investigated. The daily product sales might be the best candidates for SIMCA as they give potential to daily monitoring of sales. The daily product sales model was built only using samples from 2016. The samples from 2017 were then projected on that model and classified using SIMCA. When analyzing the model containing sales from 2016/2017, what seemed like a change in purchase behavior appeared. There was a distinct change in purchase behavior, as in figure 4.4. The SIMCA should catch such changes for it to be useful. All samples reflecting the new behavior were marked, as seen in figure 4.20.

A histogram comparing the SIMCA with the marked samples can be seen in figure 4.21. Samples not classified as a part of the model by SIMCA and the marked samples have a value of one, while the others have a value of zero. The histogram shows the sum of both, and a value of two indicates that the SIMCA classified a marked sample as unfit for the 2016 model.

The histogram shows that the marked samples (orange bars) were mostly at the end of 2017. The end of 2017 was also the period were the SIMCA found the least amount of fitting samples. More precisely, out of the 79 marked samples in 4.20 46 was also classified as non-fitting to the model by SIMCA. The SIMCA classified a total of 85 non-fitting samples in 2017. Some of them, as seen in 4.21, were the dates around Chinese new year while others were spread out across the rest of the year.

**Figure 4.20** Scores - 2016/2017 model with some 2017 dates marked

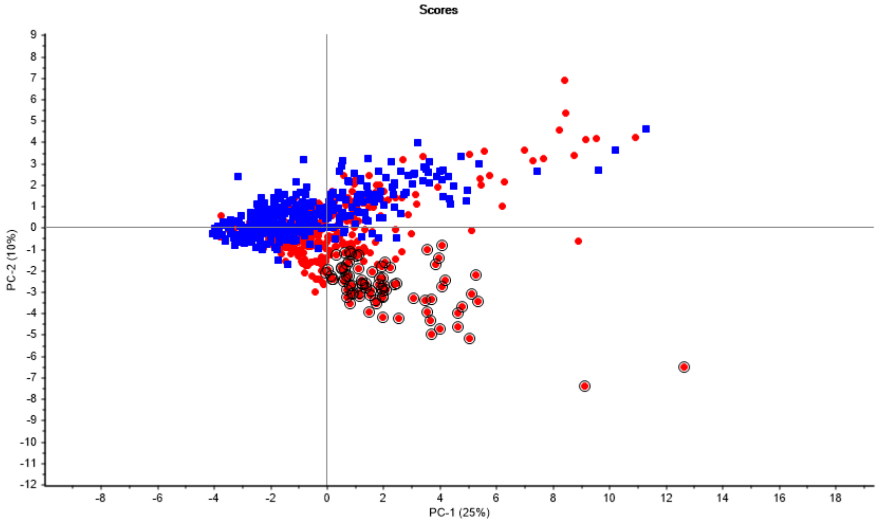
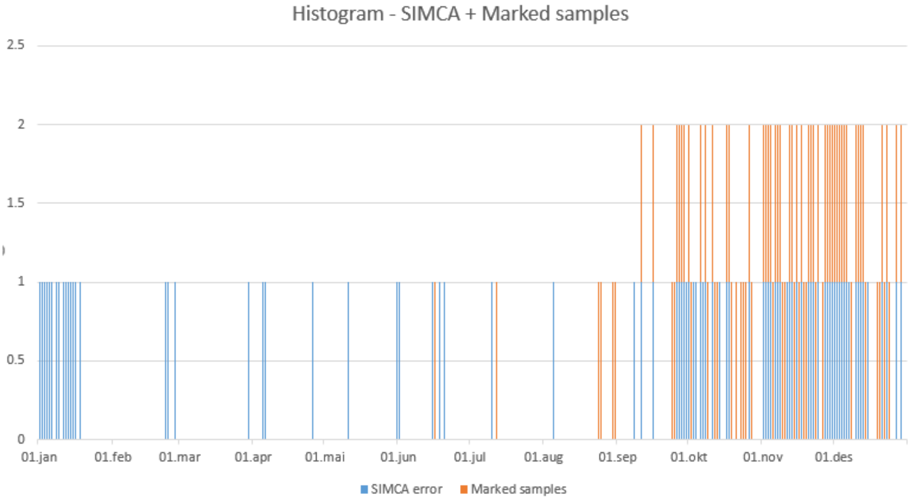


**Figure 4.21** Histogram - SIMCA model and marked samples.



### 4.7.2 Standardization of data with SIMCA

Z-score standardization has been applied to prevent popular provinces/products from gaining to much influence in the model. Standardization made investigating properties simpler, as each province/product had equal importance in the model. However, for SIMCA, one can argue that it is desirable to keep the effect of popular variables having more influence in the model. If some products popularity drops, this could be useful information. The test was again, but this time the data was not standardized, only mean-centered. The scores and loadings will not be analyzed as in the previous sections, but some remarks will be made. In figure 4.22 one can see the correlation loadings of the daily product sales without z-score standardization. In comparison to the correlation loadings when the same data has been standardized, as seen in figure 4.2, there are fewer variables with high loading values. The loadings in figure 4.2 are more spread out than the ones in figure 4.22. The scores, seen in figure 4.23, for the non-standardized data looks similar to the standardized 4.4. However, it captures a somewhat different structure. Looking at the correlation loadings, it is clear that the marked samples are a consequence of more people buying more of product number 7. In the standardized loadings (figure 4.2) the effect is more complex, consisting of a combination of different products. The explained variance of the non-standardized model can be seen in table 4.17. The explained variance is higher than the one of the standardized model 4.2. This is because in modeling the most significant variables better, the total residuals becomes smaller.

**Table 4.17:** Explained variance - Non-standardized daily product sales

| | PC-0 | PC-1 | PC-2 | PC-3 | PC-4 | PC-5 | PC-6 | PC-7 |
|---|---|---|---|---|---|---|---|---|
| Calibration | 0 | 24,93 | 34,59 | 39,43 | 43,81 | 47,99 | 51,90 | 55,58 |
| Validation | 0 | 20,84 | 27,91 | 29,25 | 30,14 | 30,83 | 32,33 | 34,20 |

In figure 4.24 one can see the same histogram as in figure 4.21. The areas marked in figure 4.20 and figure 4.23 may look similar, but they are based on two different models. Still one has similar traits in the histograms. Without standardization, the SIMCA finds 131 samples from 2017 which does not fit the model from 2016. Of the marked samples in figure 4.23 most of them were either in January (around new year) and in November/December. There was a total of 56 samples marked in the score plot, and all but 6 of them were categorized as outliers by the SIMCA model. Of the non-standardized and standardized models, the standardized model is less sensitive. This is because the non-standardized model has fewer products spanning the components.

**Figure 4.22** Loadings - 2016/2017 without standardization

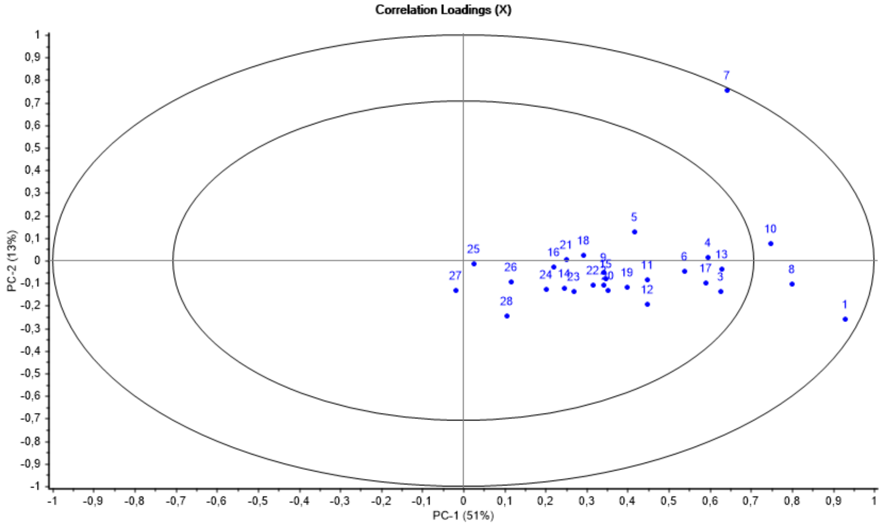


**Figure 4.23** Scores - 2016/2017 without standardization. Marked samples as in 4.20

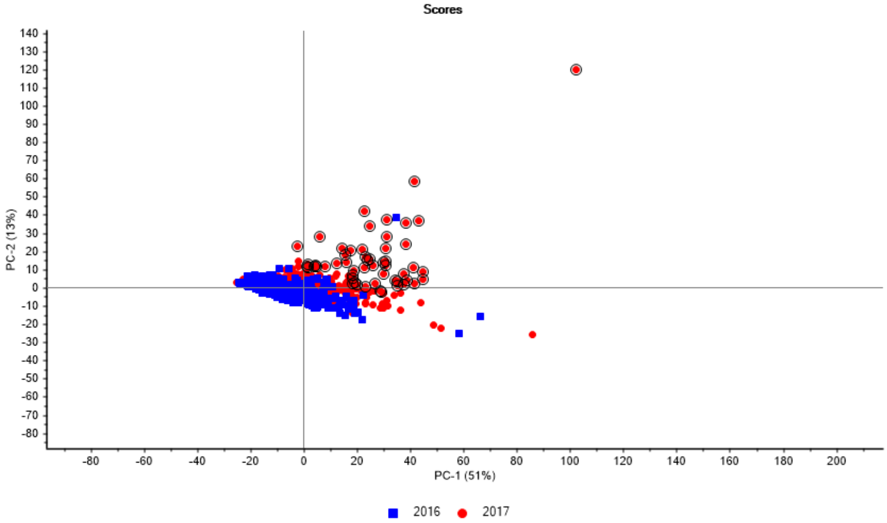
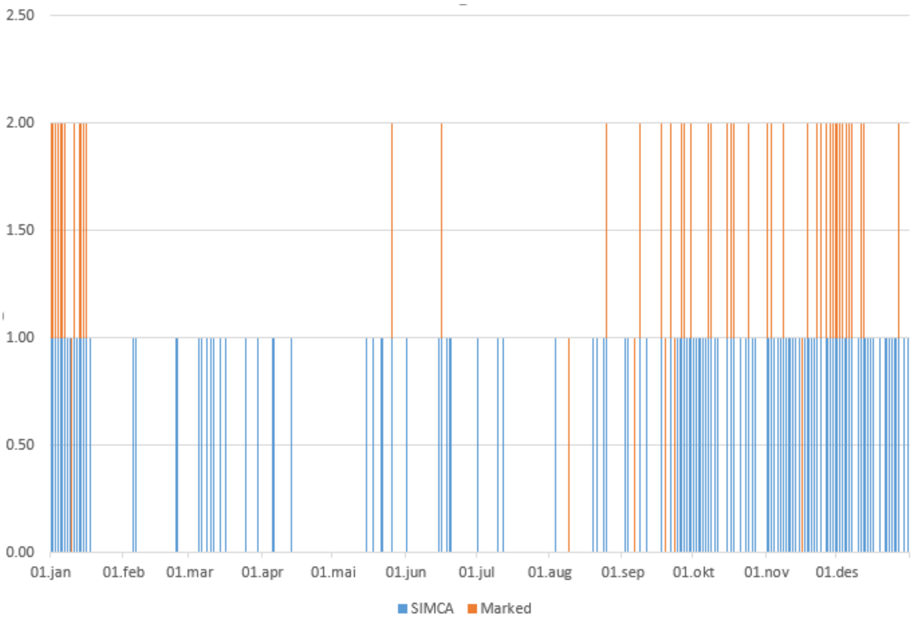


**Figure 4.24** Histogram - SIMCA on non-standardized data compared with samples marked in 4.23



Chapter 5

# Conclusion

## 5.1 Discussion on results

### 5.1.1 Daily sales

The daily sales of products/provinces both had similar results, which indicated seasonal trends and yearly changes in the company's product portfolio. The plots unveiled trends such as the effect of promotion days and seasonal changes. The yearly differences between the favorite products were also analyzed. This long-term analysis showed how some products sold worse one year than the previous. A SIMCA analysis, where samples from 2017 were tested on a model from 2016, showed how this change in purchase behavior could be detected real-time by daily projecting samples on an existing model. An advantage one has here is that by knowing the model, one can quickly analyze not just that something has changed in purchase behavior, but precisely what has changed. Investigations were made on seasonal effects, which revealed which seasons were the most popular, and also to a degree which products were mostly sold in the different seasons. This information might not be new for a company, given that it already knows the use of their products, but it can be used to confirm/debunk what they believe to be the existing purchase pattern. The daily province sales were not as interpretable as the daily product sales. Only the first principal component modeled the behavior, with the following components adding little to none, explained variance. However, as with the daily product sales, a SIMCA analysis could prove useful in detecting changes in province purchase behavior.

### 5.1.2 Purchase times

The PCA of purchase times did not reveal any surprises. The popular products were shown in the loadings and popular purchase times in the scores. These results could have been found by summing the total sales within the different intervals. However, one then misses the analytic advantage of PCA. With PCA, one can detect if there are different purchase time between different products. This could now have been revealed by summing all sales. The PCA also indicates how common this structure is, which can be interpreted from the explained variance.

### 5.1.3 Customer-Product Matrix

PCA did not perform well with the Customer-Product Matrix. This might be a consequence of that the Customer-Product Matrix was very sparse, with many zero-values. Extracting common variance in such a dataset proved difficult, and network-approach as shown in [24] yielded better results. Customer-Product Matrix and similar data structures such as, e.g., basket-case are not suitable for PCA.

## 5.2 PCA and data mining

In this thesis, the potential of PCA as a tool in data mining not just for data reduction has been examined. The results show that PCA does contain an analytic value if one interprets the scores and loadings. Promotion-dates was discovered, as well as product groupings and popular products/purchase times and high-buying products. If the PCA achieves a high explained variance, this could indicate that some strong variable relations were found. Another property of the PCA is the detection of outliers. Several promotion dates and dates with abnormal behavior were discovered when investigating how well each sample fit the model. However, the principal components also do have a rightful place as a method for data reduction. E.g., the purchase times could be 80% explained by only one component.

## 5.3 Further work

### 5.3.1 Comparison of other modeling algorithms

In this thesis, the use of PCA has been investigated as something more as data reduction within the field of data mining. However, one has not compared PCA to other modeling methods. No investigation into which modeling methods are commonly used for modeling sales records has been done. This is a weakness in this thesis, as it fails to set the modeling power of PCA into perspective.

### 5.3.2 Purchase devices

The dataset contains information about which device that was used to make the purchases. One possibility could be to make different models, one for each different device. This could then again be used to investigate differences between the different devices. The models could unveil different behaviors between the different devices which is interesting for targeted marketing. Different devices could have differences in which products are most commonly bought when they are bought and which province that use them the most.

### 5.3.3 Gender

Gender difference can also be subject to investigation. Building different models for male/female might reveal different purchase pattern. These results could be used to target advertisement.

### 5.3.4 Predicting repurchases

One other possibility is to try to predict if a customer will purchase more in the future. Combining data such as previously bought products, location, and purchase device one could use logistic regression or other methods to build a model for prediction.

### 5.3.5 SIMCA - Further testing and development

The use of SIMCA could be further investigated. For instance, SIMCA could be implemented and used in a dashboard. A dashboard, in business intelligence, is the visualization of results/models that can be directly used in decision making.

# Bibliography

[1] Nist/sematech e-handbook of statistical methods. https://www.itl.nist.gov/div898/handbook/pmc/section5/pmc543.htm. Accessed: 03-06-2018.

[2] Python data analysis library. https://pandas.pydata.org/. Accessed: 03-06-2018.

[3] Selfnutritiondata. `http://nutritiondata.self.com/facts/fruits-and-fruit-juices/1846/2`. Accessed: 05-12-2017.

[4] Chinese new year. https://en.wikipedia.org/wiki/Chinese_New_Year, May 2018. Accessed: 03-06-2018.

[5] B. K. Alsberg. *Chemometrics*. Unpublished, 0.56 edition, 2017.

[6] I. Ayres. *Super Crunchers: Why Thinking-by-numbers is the New Way to be Smart*. Bantam Books, 2008.

[7] CAMO. The unscrambler x method references. http://www.camo.com/helpdocs/The_Unscrambler_Method_References.pdf. Accessed: 01-10-2017.

[8] CAMO. The unscrambler x product page. http://www.camo.com/rt/Products/Unscrambler/unscrambler.html. Accessed: 01-10-2017.

[9] CAMO. The unscrambler x user manual. http://www.camo.com/files/TheUnscramblerXv10.3-UserManual.zip. Accessed: 01-10-2017.

[10] M. W. A. Fisher, Ronald Aylmer. 032: Studies in crop variation. ii. the manurial response of different potato varieties. pages 311–320, 1923.

[11] K. P. F.R.S. Liii. on lines and planes of closest fit to systems of points in space. *The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science*, 2(11):559–572, 1901.

[12] J. Ginsberg, M. H. Mohebbi, R. S. Patel, L. Brammer, M. S. Smolinski, and L. Brilliant. Detecting influenza epidemics using search engine query data, Nov 2008.

[13] B. Haas. Chinese shoppers spend a record $25bn in singles day splurge, Nov 2017. Accessed: 03-06-2018.

[14] J. Han, J. Pei, and M. Kamber. *Data mining: concepts and techniques*. Elsevier, 2011.

[15] Harald and M. Martens. *Multivariate Analysis of Quality - An Introduction*. John Wiley sons, LTD, 2001.

[16] T. Heggernes. *Digital forretningsforståelse*. Fagbokforlaget, 2 edition, 2017.

[17] H. Hotelling. Analysis of a complex of statistical varaible into principal components. *Journal of Educational Psychology, 24*, pages 417–441 and 498–520, 1993.

[18] J. Inc. `http://corporate.jd.com/aboutUs`. Accessed: 03-06-2018.

[19] IPC. State of e-commerce: global outlook 2016-21. `https://www.ipc.be/en/knowledge-centre/e-commerce/articles/global-ecommerce-figures-2017`. Accessed: 03-06-2018.

[20] H. K. Kim, J. K. Kim, and Q. Y. Chen. A product network analysis for extending the market basket analysis. *Expert Systems with Applications*, 39(8):7403 – 7410, 2012.

[21] S. Levitt and S. Dubner. *SuperFreakonomics: Global Cooling, Patriotic Prostitutes and Why Suicide Bombers Should Buy Life Insurance*. HarperCollins Canada, 2011.

[22] Nicoguaro. Gaussianscatterpca. https://commons.wikimedia.org/wiki/File:GaussianScatterPCA.s Accessed: 03-06-2018.

[23] Y. Tominaga. Comparative study of class data analysis with pca-lda, simca, pls, anns, and k-nn, Aug 1999.

[24] H. Wang and S. Wang. A knowledge management approach to data mining process for business intelligence, Feb 2008.

[25] F. Westad, M. Hersleth, P. Lea, and H. Martens. Variable selection in pca in sensory descriptive and consumer data, Apr 2003.

[26] I. H. Witten, E. Frank, M. A. Hall, and C. J. Pal. *Data Mining: Practical machine learning tools and techniques*. Morgan Kaufmann, 2016.

[27] H. Wold. Nonlinear estimation by iterative least squares procedures, in f. david (editor), research papers in statistics. pages 411–444, 1996.

[28] S. Wold, K. Esbensen, and P. Geladi. Principal component analysis. *Chemometrics and Intelligent Laboratory Systems*, 2(1):37 – 52, 1987. Proceedings of the Multivariate Statistical Workshop for Geologists and Geochemists.

# Appendix

```
def dailyProductProvinceSales(data):
    #This function transforms the dataset to the daily province/product sales
     ↪  form.
    #Both products and sales are here set as columns. Must be separated
     ↪  afterwards.

    #Create empty data matrix.

    product_ids = []
    for item in data.Product_ID:
        if item not in product_ids:
            product_ids.append(item)

    buy_dates = []
    for buy_date in data.Order_time:
        t = pd.Timestamp(buy_date.year,buy_date.month, buy_date.day)
        if t not in buy_dates:
            buy_dates.append(t)

    cur_week = 1
    for date in buy_dates:
        if cur_week == 1:
            if date.day > 7 :
                cur_week += 1

    addresses =  []
    for address in data.Customer_address:
        if address[:2] not in addresses:
            addresses.append(address[:2])

    data1 = pd.DataFrame(index = buy_dates, columns = [product_ids])
    data2 = pd.DataFrame(index = buy_dates, columns = [addresses])
    newdata = pd.concat([data1, data2],  axis=1, join_axes=[data1.index])

    newdata = newdata.fillna(0)

    #Get first day
    newdata[data.Product_ID[0]][buy_dates[0]] += data.Quantity[0]
    order_time_iterator = 0              #To keep track of which day where at
    for i in range (1,len(data) - 1):
        if data.Order_time[i].day != data.Order_time[i-1].day:
            order_time_iterator += 1
            print(order_time_iterator, i)
        newdata[data.Product_ID[i]][buy_dates[order_time_iterator]] +=
         ↪  data.Quantity[i]
        newdata[data.Customer_address[i][:2]][buy_dates[order_time_iterator]] +=
         ↪   data.Quantity[i]


    return newdata
```

```
def productPurchaseTimes(dataset):
    #Product purchase times function.
    #Sums all sales into half-hour intervals.

    product_ids = []
    for item in dataset.Product_ID:
        if item not in product_ids:
            product_ids.append(item)

    timestamps = []

    half_hour = 0;
    hour = 0;
    for i in range (0,48):
        time = pd.Timestamp(2000,1,1,hour,half_hour)
        timestamps.append(time)

        if half_hour == 30:
            half_hour = 0
            hour += 1
        else:
            half_hour = 30


    prod_buy_times = pd.DataFrame(index = timestamps, columns = [product_ids])
    prod_buy_times = prod_buy_times.fillna(0)

    for i in range (0, len(dataset) - 1):
        hour = dataset.iloc[i].Order_time.hour
        if dataset.iloc[i].Order_time.minute >= 30:
            minute = 30
        else:
            minute = 0


        time = pd.Timestamp(2000,1,1,hour,minute)
        prod_buy_times[dataset.Product_ID[i]][time] +=
         ↪   dataset.iloc[i].Quantity

    return prod_buy_times
```

```
def provincePurchaseTimes(dataset):
    #Province purchase times function.
    #Sums all sales into half-hour intervals.
    addresses =  []
    for address in dataset.Customer_address:
        if address[:2] not in addresses:
            addresses.append(address[:2])

    timestamps = []
    half_hour = 0;
    hour = 0;
    for i in range (0,48):
        time = pd.Timestamp(2000,1,1,hour,half_hour)
        timestamps.append(time)

        if half_hour == 30:
            half_hour = 0
            hour += 1
        else:
            half_hour = 30


    newdata = pd.DataFrame(index = timestamps, columns = [addresses])
    newdata = newdata.fillna(0)

    for i in range (0, len(dataset) - 1):
        hour = dataset.iloc[i].Order_time.hour
        if dataset.iloc[i].Order_time.minute >= 30:
            minute = 30
        else:
            minute = 0

        address = dataset.iloc[i].Customer_address[:2]

        time = pd.Timestamp(2000,1,1,hour,minute)
        newdata[address][time] += dataset.iloc[i].Quantity


    return newdata
```

```
def productCustomerMatrix(dataset):
    #Function to extract the customer-product matrix.
    customers = []

    for customer in dataset.Account_name:
       if customer not in customers:
           customers.append(customer)

    products = []
    for product in dataset.Product_ID:
        if product not in products:
            products.append(product)

    products.append('nBuys')
    products.append('lastBuyTime')
    customer_product_matrix = pd.DataFrame(index = customers, columns =
     ↪   products)
    customer_product_matrix.lastBuyTime =
     ↪   pd.to_datetime(customer_product_matrix.lastBuyTime)
    customer_product_matrix = customer_product_matrix.fillna(0)


    for i in range (0, len(dataset)):

         ↪   customer_product_matrix.loc[dataset.loc[i,'Account_name'],dataset.loc[i,'Product_ID']]
         ↪   += dataset.loc[i,'Quantity']
        if
         ↪   customer_product_matrix.loc[dataset.loc[i,'Account_name'],'lastBuyTime']
         ↪   != dataset.loc[i,'Order_time']:

             ↪   customer_product_matrix.loc[dataset.loc[i,'Account_name'],'lastBuyTime']
             ↪   = dataset.loc[i,'Order_time']
            customer_product_matrix.loc[dataset.loc[i,'Account_name'],'nBuys']
             ↪   += 1
    return customer_product_matrix
```