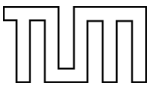


Technische Universität München

Department of Mathematics

Master's Thesis

Statistical Analysis of the Number of Sales of a Pharmacy Product

Özge Sahin

Supervisor: Prof. Claudia Czado, PhD
Advisors: Prof. Claudia Czado, PhD
Alexander Kreuzer
Submission Date: 16.10.2019

I assure the single handed composition of this master's thesis only supported by declared resources.

Garching,

Acknowledgements

First and foremost, I would like to express my gratitude to Prof. Claudia Czado for her guidance throughout the thesis process. I am deeply indebted to her for the suggestions and clarity which she provided me with.

In the same way I am thankful to Alexander Kreuzer for his helpful suggestions, the amount of time and the patience committed to this project.

Furthermore, I thank SciScry GmbH for providing me with the data which were necessary for the realization of this thesis and inspiring me with their valuable comments. Especially Philipp Beer and Piero Ferrarese have invested many hours into this work and were always there to help and improve me.

Beyond this, I am grateful to Alper, Begüm, Ekin, Emre, Gizem, Marija, Nazlican, Sofia, Tanay and all my other friends for their friendships and unconditional support. Ekin, thank you for being such a great supporter from abroad. Marija and Emre, thank you for the many encouraging discussions especially during the last months of completing my thesis. Tanay, thank you for sharing all ups and downs of my studies.

Finally and most importantly, I would like to thank my family and my brother for their encouragement and love throughout my life. Your constant support constituted the foundation for my personal development.

Abstract

As the amount of data has increased more and more over the years, the statistical analysis of the data has gained much more importance. Each data set needs a special model and the interpretability of the results is crucial for data owners.

We provide a statistical analysis of the number of sales of a pharmacy product, which investigates the explanatory power of the Google trends, the temperature and the state holidays data in Germany together with the harmonic regression components. We find that using the Google trends data as an explanatory variable usually leads to a significant improvement in the forecasting performance. Based on the model fitting results, we perform a clustering. Lastly, we present a first dependence analysis using the copula approach and show that a clustering approach that models the dependence between the clusters is needed.

Zusammenfassung

Da die Datenmenge im Laufe der Jahre immer mehr gestiegen ist, hat die statistische Analyse der Daten an Bedeutung gewonnen. Jeder Datensatz benötigt einen speziellen Modellansatz. Außerdem ist die Interpretierbarkeit der Modellergebnisse für den Datenbesitzer entscheidend.

Wir entwickeln eine statistische Analyse eines Apothekenproduktverkaufs, der die Aussagekraft von Google Trends, der Temperatur und Feiertagsdaten in Deutschland sowie der harmonische Regressionskomponenten untersucht. Wir stellen fest, dass die Verwendung der Google Trendsdaten als erklärende Variable in den meisten Regionen zu einer signifikanten Verbesserung der Prognose führt. Basierend auf den Ergebnissen der Modellanpassung führen wir ein Clustering durch. Schließlich präsentieren wir eine erste Abhängigkeitsanalyse mit Hilfe eines Kopula-Ansatz und zeigen, dass ein Clustering-Ansatz, der die Abhängigkeit zwischen den Clustern modelliert, erforderlich ist.

Contents

1 Introduction **1**

2 Theoretical background **3**
2.1 Important distributions . . . 3
2.2 Data preprocessing . . . 6
2.2.1 Outlier detection and replacement . . . 6
2.2.2 Data transformation . . . 8
2.2.3 Data splitting . . . 8
2.3 Linear regression . . . 10
2.3.1 Parameter estimation: β and σ . . . 11
2.3.2 Hypothesis testing . . . 11
2.3.3 Subset selection . . . 12
2.3.4 Diagnostics and model evaluation . . . 13
2.4 Harmonic regression . . . 15
2.4.1 Parameter estimation: ω . . . 16
2.4.2 Parameter estimation: α and β . . . 21
2.5 Performance measures for comparing statistical models . . . 22
2.6 Clustering . . . 24
2.6.1 Overview of clustering methods . . . 25
2.6.2 Distance-based clustering methods . . . 26
2.6.3 Validation indices . . . 31
2.7 Copulas . . . 34
2.7.1 Definition and basic properties . . . 34
2.7.2 Bivariate copula families . . . 35
2.7.3 Dependence measures . . . 38
2.7.4 Exploratory visualization . . . 41
2.7.5 Bivariate parametric copula estimation . . . 43

3 Data set **45**
3.1 Introduction to data set . . . 45
3.2 Third party data . . . 46
3.3 Representative selection . . . 48
3.4 Data preprocessing steps . . . 48

4 Marginal model fitting **50**

4.1 Marginal model specification 50
4.2 Marginal models comparison 56

5 Exploration of spatial dependence patterns 58

6 Clustering 61
6.1 Feature selection 61
6.2 Distance-based clustering analysis 65
6.2.1 1st scenario: Mean of the time series data sets as all features 67
6.2.2 2nd scenario: Mean and standard deviation of time series data sets as all features 68
6.2.3 Comparison of the scenarios 69
6.2.4 The exploration of the best clustering, `pl` region segmentation . . . 70

7 Dependence modelling 74
7.1 u-data 74
7.2 Pairs plots 79

8 Conclusion 93

Chapter 1

Introduction

A pharmacy software company provides pharmacies with a system that stores data and maintains product use process within the pharmacies. The system additionally includes security limits to protect patient health data. A patient identity is anonymous; however, the date, location and sold products from a patient transaction are recorded. To know in which regions these transactions are showing similar behavior, helps the software company to improve its product. For instance, an inventory management tool for the targeted pharmacies can be offered thereby reducing their inventory costs.

In order to find the groups with the similar behavior in the data, a statistical model is needed. Bloomfield (2000) discussed the harmonic regression model to study the periodic behavior. Durbin (1960) justified the use of the linear regression model to find the coefficients of the harmonic regression model. It additionally helps to have both third party data and the harmonic regression components in the same model.

Analyzing the result of the statistical model components to detect similar groups in the data can be grouped into clustering analysis. There are many clustering methods such as k-means and hierarchical clustering available. Even though they are not appropriate when the groups have different dependence and variability patterns, their popularity has been continuing due to their interpretability.

In the thesis, we will analyze the number of sales of a pharmacy product. As a third party data, we will make use of the Google trends data, the temperature data and the state holidays data in Germany for the marginal model fitting. Based on the harmonic regression model components, we will apply clustering. Because our clustering approaches do not account for the dependence between the clusters, we will perform the dependence analysis with the copula approach, based on the work of Sklar (1959).

The remainder of this thesis is organized as follows. Chapter 2 gives a review of the statistical concepts our analysis builds on. In Chapter 3, we introduce the data set that the company provided us and the third party data sets that we obtained from several sources. Chapter 4 deals with the harmonic regression model fits. In Chapter 5, we put

the model results and some third party data sets together and analyze their relationships. Chapter 6 provides a clustering of the Germany regions for the company. In Chapter 7, we give a visual and numerical analysis of the dependence structure within the areas. Chapter 8 concludes.

Chapter 2

Theoretical background

In this chapter we introduce mathematical concepts and definitions we use in the course of this thesis. Starting with the definitions of some univariate and multivariate distributions, we describe the main concepts of data analysis like data preprocessing and performance measures. Immediately afterwards we explain our models that we fit to our data and their backgrounds. We conclude this chapter with the essential ideas of copulas, which we define in the Section 2.7.

Throughout the chapter we denote random variables with capital letters and observed values with small letters. Moreover, we assume the continuity of all random variables.

2.1 Important distributions

We start with the definitions and properties of the probability distributions we apply in the thesis. Let f stand for the probability density function (pdf) and F the cumulative distribution function (cdf). For this section, we consult Czado (2019).

Uniform distribution

Definition 2.1.1. *(Probability density function of uniform distribution)*
*Let $X \in \mathbb{R}$ be a random variable following the **uniform distribution**. Then the probability density function (pdf) of X at x is defined as*

$$f(x) := \begin{cases} \frac{1}{b-a} & for \;\; a \leq x \leq b \\ 0 & for \;\; a < x \;\; or \;\; x > b. \end{cases} \tag{2.1.1}$$

And we denote

$$X \sim \mathcal{U}(a, b).$$

If $a = 0$ and $b = 1$, i.e. $X \sim \mathcal{U}(0, 1)$, X follows a standard uniform distribution.

Normal distribution

Definition 2.1.2. *(Probability density function of univariate normal distribution)* *Let $X \in \mathbb{R}$ be a random variable following the* ***univariate normal distribution*** *with mean μ and variance $\sigma^2 > 0$. Then the probability density function of X at x is defined as*

$$f(x; \mu, \sigma^2) := \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left\{-\frac{1}{\sigma^2}(x - \mu)^2\right\}. \tag{2.1.2}$$

And we denote

$$X \sim \mathcal{N}(\mu, \sigma^2).$$

If $\mu = 0$ and $\sigma^2 = 1$, i.e. $X \sim \mathcal{N}(0, 1)$, X follows a standard normal distribution and its probability density function is represented by ϕ, whereas its cumulative distribution function by Φ.

Definition 2.1.3. *(Probability density function of multivariate normal distribution)* *For the multivariate case, let $\boldsymbol{X} = (X_1, ..., X_d)^\top \in \mathbb{R}^d$ be a random vector having the* ***multivariate normal distribution*** *with mean vector $\boldsymbol{\mu} = (\mu_1, ..., \mu_d)^\top \in \mathbb{R}^d$ and the positive covariance matrix $\Sigma = (\sigma_{ij})_{i,j=1,...,d} \in \mathbb{R}^{d\times d}$. Then the probability density function of $\boldsymbol{X}$ at $\boldsymbol{x}$ is given by*

$$f(\boldsymbol{x}; \boldsymbol{\mu}, \Sigma) := \frac{1}{(2\pi)^{d/2}} det(\Sigma)^{-1/2} \exp\left\{-\frac{1}{2}(\boldsymbol{x} - \boldsymbol{\mu})^\top \Sigma^{-1}(\boldsymbol{x} - \boldsymbol{\mu})\right\}. \tag{2.1.3}$$

And we write

$$\boldsymbol{X} \sim \mathcal{N}_d(\boldsymbol{\mu}, \Sigma).$$

Student's t-distribution

Definition 2.1.4. *(Probability density function of univariate Student's t-distribution)* *Let $X \in \mathbb{R}$ be a random variable following the* ***univariate Student's t-distribution*** *with mean $\mu \in \mathbb{R}$, scale parameter $\sigma^2 > 0$ and degrees of freedom $\nu > 0$. Then the probability density function of X at x is defined as*

$$f_\nu(x; \mu, \sigma^2) := \frac{\Gamma(\frac{\nu+1}{2})}{\Gamma(\frac{\nu}{2})\sqrt{\pi\nu}\sigma} \left(1 + \left(\frac{x - \mu}{\sigma}\right)^2 \frac{1}{\nu}\right)^{-\frac{\nu+1}{2}}. \tag{2.1.4}$$

And we write

$$X \sim t_\nu(\mu, \sigma^2).$$

If $\mu = 0$ and $\sigma^2 = 1$, X follows **univariate standard Student's t-distribution** and its cumulative distribution function is represented by T_ν. For $\nu > 2$ the mean and variance are given by

$$E(X) = \mu \quad \text{and} \quad Var(X) = \frac{\nu}{\nu - 2}\sigma^2. \tag{2.1.5}$$

Definition 2.1.5. *(Probability density function of multivariate Student's t-distribution) For the multivariate case, following Kotz et al. (1995), let* $\boldsymbol{X} = (X_1, ..., X_d)^\top \in \mathbb{R}^d$ *be a random vector having the* ***multivariate Student's t-distribution*** *with mean vector* $\boldsymbol{\mu} = (\mu_1, ..., \mu_d)^\top \in \mathbb{R}^d$*, symmetric and positive-definite scale parameter matrix* $\Sigma \in \mathbb{R}^{d\times d}$ *with unit diagonal entries and elements* ρ_{ij} *for* $i, j = 1, ..., d$ *and* $\nu > 0$ *degrees of freedom. Then the probability density function of* $\boldsymbol{X}$ *at* $\boldsymbol{x}$ *is given by*

$$f_\nu(\boldsymbol{x}; \boldsymbol{\mu}, \Sigma) := \frac{\Gamma(\frac{\nu+d}{2})}{\Gamma(\frac{\nu}{2})(\pi\nu)^{\frac{d}{2}}} det(\Sigma)^{-1/2} \left(1 + \frac{1}{\nu}(\boldsymbol{x} - \boldsymbol{\mu})^\top \Sigma^{-1} (\boldsymbol{x} - \boldsymbol{\mu})\right)^{-\frac{\nu+d}{2}}. \tag{2.1.6}$$

And we write

$$\boldsymbol{X} \sim t_d(\nu, \boldsymbol{\mu}, \Sigma).$$

Here the matrix Σ is the correlation matrix of $\boldsymbol{X}$, i.e. $Cor(X_j, X_k) = \rho_{jk}$. The Student's t-distribution is said to be central if $\boldsymbol{\mu} = \boldsymbol{0}$, otherwise non-central.

Definition 2.1.6. *(Probability density function of univariate skew-t distribution) Following Azzalini et al. (2003), let* $X \in \mathbb{R}$ *be a random variable having the* ***univariate skew-t distribution*** *with parameters* ξ*,* ω *and* α*, which regulate location, scale and skewness* ($\omega > 0$)*, respectively, and with degrees of freedom* $\nu > 0$*. Then the probability density function of* X *at* x *is defined as*

$$f_\nu(x; \xi, \omega, \alpha) := \frac{2}{\omega} t_\nu(z) T_{\nu+1}\left(\alpha z \sqrt{\frac{\nu+1}{\nu+z^2}}\right), \quad z = \frac{x-\xi}{\omega}, \tag{2.1.7}$$

where t_ν *denotes the probability density function of the univariate Student's t-distribution with* ν *degrees of freedom and* $T_{\nu+1}$ *is the cumulative distribution function of the univariate Student's t-distribution with* $\nu + 1$ *degrees of freedom.*

And we write

$$X \sim ST(\xi, \omega, \alpha, \nu).$$

If $\alpha = 0$, the univariate skew-t distribution reduces to the univariate Student's t-distribution.

2.2 Data preprocessing

After introducing some often utilized statistical distributions and tests, we now describe data preprocessing steps we follow in the thesis. Data preprocessing is extremely important, because it allows us to remove the effects of outlying and non plausible data. In the following subsections we clarify the steps being of data cleaning, e.g. identify outliers and replace them, data transformation, as well as data splitting.

2.2.1 Outlier detection and replacement

An outlier in a data set can be delineated as an observation or a set of observations which appears to be inconsistent with the data set as stated by Barnett et al. (1994). Detecting outliers is critical as they influence the model selection, parameter estimation and, consequently, forecasts. There are many techniques for outlier detection in the literature; however, we use the approach by Hyndman et al. (2019) as implemented in `R` package `forecast`. We introduce first the LOESS curve fitting approach and moving average model since they are utilized by the outlier detection technique we use.

LOESS Curve Fitting (Local regression)

Local regression is used to model a relationship between a known variable, called **predictor, explanatory or independent variable**, and a random variable, called **response, outcome or dependent variable**, which is related to the predictor variable. Assume a model of the form

$$Y_j = f(x_j) + \epsilon_j \quad \text{for } \ j = 1, ..., n, \tag{2.2.1}$$

where f is an arbitrary unknown function and ϵ_j is an independently and identically distributed (i.i.d) random error with $E(\epsilon_j) = 0$ and finite variance $E(\epsilon_j)^2 = \sigma^2 < \infty$ for $j = 1, ..., n$. We aim to obtain the estimate $\hat{f}(x_0)$ around a target covariate x_0.

In local linear regression an optimal $\hat{f}(x_0)$ minimizes the expected squared error of the estimated function and depends highly on the *smoothing window* $[x_0 - h(x_0), x_0 + h(x_0)]$ with bandwidth $h(x_0)$. To estimate $f(x_0)$, only observations within this window are used. Furthermore, the observations are weighted by the weight sequence

$$w_j(x_0) := W\left(\frac{x_j - x_0}{h(x_0)}\right) \quad \text{for } \ j = 1, ..., n, \tag{2.2.2}$$

where $W(.)$ is a weight function assigning smallest weights to the observations far away from x_0. Tukey's tri-weight function given in Equation (2.2.3) is an example for it.

$$W(x) := \begin{cases} (1 - |x|^3)^3 & for \ \ |x| \leq 1 \\ 0 & for \ \ |x| > 1. \end{cases} \tag{2.2.3}$$

While it is common to apply local linear regression with a window size that is constant over the range of the predictor variable, there are other methods based on locally adaptive

bandwidths like super smoother, which are proven to be superior in some cases such as heteroskedastic error variance as can be seen in Friedman (1984). Moreover, the super smoother approach is implemented by the `R` package `forecast`. For more details about this subsection, we refer to Loader (1999).

Moving - average model (MA model)

Definition 2.2.1. *(MA model (q))*
Let X_t denote a time series for $t = 1, ..., T$. Then the ***MA(q) model*** *is defined as*

$$X_t := \mu + \sum_{j=1}^{q} b_j \epsilon_{t-j} + \epsilon_t, \tag{2.2.4}$$

where μ is the expectation of X_t, $b_1, ..., b_q$ are the MA-parameters with $b_q \neq 0$ and the error term ϵ_t is a series of uncorrelated random variables with $E(\epsilon_t) = 0$ and finite variance $E(\epsilon_t^2) = \sigma^2 < \infty$.

Now, we explain how to detect outliers by using LOESS curve fitting and MA model.

Outlier detection

We can have either non-seasonal or seasonal data, in which we want to identify potential outliers. By seasonality, we mean periodic fluctuations.

For the non-seasonal data, first of all, residuals are calculated by fitting a LOESS curve with Friedman's super smoother as shown in Friedman (1984) and discussed above. If the residuals lie outside the range, denoted by int,

$$int = \pm 3.0(q_{0.75} - q_{0.25}), \tag{2.2.5}$$

where q_p is the empirical p-quantile of the residuals, they are treated as outliers and their corresponding value is removed from the data.

For the seasonal data, firstly, the seasonal component is removed, then the same outlier detection approach described for the non-seasonal data can be applied to the remainder component.

Outlier replacement

Until now, we have described how to identify outliers in the data set. Now we discuss **linear interpolation**, which is the technique we use for the replacement of the outliers discussed in Hyndman et al. (2013).

Assume, without loss of generality (w.l.o.g), that $\boldsymbol{X} = (x_1, ..., x_d)$ are data points in ascending order, that is $x_j < x_{j+1}$ for $j = 1, ..., (d-1)$, and $\boldsymbol{Y} = (y_1, ..., y_d)$ are their corresponding values. Let x be a point such that $x_j < x < x_{j+1}$. Then the linear

interpolation value for y at point x is given by

$$\hat{y}_x := y_j + \frac{(y_{j+1} - y_j)(x - x_j)}{(x_{j+1} - x_j)} \quad \text{for } j = 1, ..., (d-1). \tag{2.2.6}$$

Hence, if the observation y_j is regarded as an outlier by our technique, we replace its value via linear interpolation, $\hat{y}_x$.

2.2.2 Data transformation

One of the important steps of the data preprocessing is data transformation. We can have large variability in the data or want to get the data conform more closely to the normal distribution. Moreover, the response variable can be always positive and we may always need to predict positive values. In such cases we may transform either predictor variables or response variable or both. The data transformation we employ in the thesis is **log-transformation** and **logit-transformation**. More details can be found in Hyndman et al. (2013).

Let $k > 0$ and j be real numbers. If there exists a real number $b > 0$ such that $b^j = k$, then we say that the **base-b logarithm** of k is j and the expression $\log_b(k)$ is used to denote the base-b logarithm function of k. In the thesis we draw on the natural logarithm, whose base number is the transcendental number e with the decimal expansion $2.718282\ldots$ Moreover, we refer to the natural logarithm with the abbreviation $\ln(\cdot)$. In other words, let $y \in \mathbb{R}^n_{++}$ be a variable and $\widetilde{y} \in \mathbb{R}^n$ be its natural logarithm transformation. Then we denote

$$\widetilde{y} := \ln(y). \tag{2.2.7}$$

For logit transformation, let $y \in (0, 1)$ be a variable and $\widetilde{y} \in \mathbb{R}$ be its natural logit transformation. Then we denote

$$\widetilde{y} := \ln\left\{\frac{y}{1-y}\right\}. \tag{2.2.8}$$

Another important data transformation is normalization. In normalization, the data is scaled into a range of $[0, 1]$ by using the following formula:

Let $\boldsymbol{Y} = (y_1, ..., y_d)$ contain d data points. Then the normalization of the data is given by

$$\boldsymbol{Y}_{norm} := \frac{\boldsymbol{Y} - \boldsymbol{Y}_{min}}{\boldsymbol{Y}_{max} - \boldsymbol{Y}_{min}}. \tag{2.2.9}$$

2.2.3 Data splitting

Another noteworthy step in the data preprocessing is data splitting, where we split the data into training and test set. With the training set we learn patterns of the data and the test set is regarded as a proxy for how the model would perform in the real world.

There are many classical statistical sampling techniques for the data splitting as discussed by May et al. (2010). However, as we are dealing with time series data, which is naturally ordered, we employ the deterministic method, so-called convenience sampling, as summarized by Reitermanova (2010).

Let $\boldsymbol{X} = \{X_1, ..., X_T\}^\top$ be a time series with $|\boldsymbol{X}| = T$ and $\boldsymbol{X}_{train}$, $\boldsymbol{X}_{test}$ denote respectively the training and test set. Furthermore, suppose that p_{train} and p_{test} stand for fraction of training and test set such that $p_{train} + p_{test} = 1$. Then we split the data set $\boldsymbol{X}$ according to

$$\boldsymbol{X}_{train} := \{X_1, ..., X_{p_{train} \cdot T}\}, \tag{2.2.10}$$

$$\boldsymbol{X}_{test} := \{X_{p_{train} \cdot T+1}, ..., X_T\}. \tag{2.2.11}$$

If $p_{train} \cdot T$ is not an integer, we round up the value. Moreover, there is not a general rule about how to choose the fractions and research about optimal splitting for supervised learning problems has been continuing as can be seen in Kevin et al. (2011). In general the complexity of the model being fit to the data and signal-to-noise ratio in the data are known to affect it as discussed by Hastie et al. (2009). Nonetheless, we apply $p_{train} = 0.7$ and $p_{test} = 0.3$ in the thesis.

2.3 Linear regression

After introducing some statistical distributions and data preprocessing steps, now we discuss the models to be fitted to our data.

We start with the linear regression, which is one of the most commonly used statistical method. It defines the relationships among the variables and its models construct a very flexible framework for describing and testing hypotheses about the relationships. For more details about this section, see for example Fox (1997).

Let $\boldsymbol{y} = (y_1, ..., y_n)^\top \in \mathbb{R}^n$ be a vector of the observed values of the random variables, Y_j for $j = 1, ..., n$, being predicted in terms of k known predictors $x_{j1}, ..., x_{jk}$. Then the **linear regression model** explains the random response $\boldsymbol{Y}$ as a linear function of the predictors

$$Y_j = \beta_0 + \beta_1 x_{j1} + ... + \beta_k x_{jk} + \epsilon_j \quad \text{for} \quad j = 1, ..., n, \tag{2.3.1}$$

where the parameters consist of the intercept β_0 and the k parameters $\beta_1, ..., \beta_k$. Further, ϵ_j's are independently and normally distributed random errors with $E(\epsilon_j) = 0$ and constant variance $E(\epsilon_j^2) = Var(\epsilon_j) = \sigma^2$.

Moreover, we define the response $\boldsymbol{Y}$, regression coefficient $\boldsymbol{\beta}$, error $\boldsymbol{\epsilon}$ and the design matrix X as

$$\begin{aligned}
\boldsymbol{Y} &= (Y_1, ..., Y_n)^\top \in \mathbb{R}^n, \\
\boldsymbol{\epsilon} &= (\epsilon_1, ..., \epsilon_n)^\top \in \mathbb{R}^n, \\
\boldsymbol{\beta} &= (\beta_0, \beta_1, ..., \beta_k)^\top \in \mathbb{R}^p, \\
X &= \begin{pmatrix} 1 & x_{11} & x_{12} & ... & x_{1k} \\ 1 & x_{21} & x_{22} & ... & x_{2k} \\ & & \vdots & & \\ 1 & x_{n1} & x_{n2} & ... & x_{nk} \end{pmatrix} \in \mathbb{R}^{n \times p}.
\end{aligned}$$

Then we can rewrite the Equation (2.3.1) as

$$\boldsymbol{Y} = X\boldsymbol{\beta} + \boldsymbol{\epsilon} \quad \text{with} \quad \boldsymbol{\epsilon} \sim \mathcal{N}_n(\boldsymbol{0}, \sigma^2 I_n), \tag{2.3.2}$$

where I_n is the $n \times n$ identity matrix and $\mathcal{N}_n(\boldsymbol{\mu}, \Sigma)$ is introduced in Equation (2.1.3). Furthermore, from Equation (2.3.2) it follows

$$E(\boldsymbol{Y}) = X\boldsymbol{\beta} \quad \text{and} \quad Var(\boldsymbol{Y}) = \sigma^2 I_n. \tag{2.3.3}$$

and under the normality assumption of the error ϵ we have

$$\boldsymbol{Y} \sim \mathcal{N}_n(X\boldsymbol{\beta}, \sigma^2 I_n). \tag{2.3.4}$$

As can be seen in the Equation (2.3.1), now we need to estimate the regression coefficient $\boldsymbol{\beta}$ and standard deviation of the error σ based on n observations.

2.3.1 Parameter estimation: β and σ

There are two main techniques utilized for the parameter estimation, least squares and maximum likelihood estimation. However, if the assumptions of independence, constant variance and normality are satisfied, both techniques yield the same estimate. Hence, we acquaint the reader with **maximum likelihood estimation**.

In general the maximum likelihood estimation benefits from the likelihood function $\mathcal{L}(\boldsymbol{\theta}; x)$. If we have a statistical model, i.e. family of distributions $\{f(.; \boldsymbol{\theta}) | \boldsymbol{\theta} \in \Theta\}$, where θ accounts for the parameters of the model and Θ for the parameter space, the maximum likelihood method estimates $\boldsymbol{\theta}$ assuming that there is a unique maximum by

$$\hat{\boldsymbol{\theta}} = \text{argmax}_{\boldsymbol{\theta} \in \Theta} \mathcal{L}(\boldsymbol{\theta}; x). \tag{2.3.5}$$

In the linear regression case, from Equation (2.3.4) the likelihood of $\boldsymbol{\theta} = (\boldsymbol{\beta}, \sigma)$ given $\boldsymbol{y}$ follows

$$\mathcal{L}(\boldsymbol{\beta}, \sigma | \boldsymbol{y}) = \frac{1}{(2\pi\sigma^2)^{n/2}} \exp\left\{-\frac{1}{2\sigma^2}\|\boldsymbol{y} - X\boldsymbol{\beta}\|^2\right\}. \tag{2.3.6}$$

Therefore, to find maximum in Equation (2.3.5), when the likelihood is given by Equation (2.3.6), we use partial differentiation and solve the resulting normal equations to get

$$\hat{\boldsymbol{\beta}} = (X^\top X)^{-1} X^\top \boldsymbol{y} \tag{2.3.7}$$

and

$$\hat{\sigma}^2 = \frac{1}{n} \sum_{j=1}^{n} (y_j - \hat{y_j})^2. \tag{2.3.8}$$

On the other hand, an unbiased estimator of σ^2 is derived from Equation (2.3.8) by

$$s^2 = \frac{1}{n-p} \sum_{j=1}^{n} (Y_j - \hat{Y}_j)^2 \tag{2.3.9}$$

$$= \frac{n}{n-p} \hat{\sigma}^2. \tag{2.3.10}$$

Additionally, under Equation (2.3.2), we can easily show that

$$\hat{\boldsymbol{\beta}} \sim \mathcal{N}_p(\boldsymbol{\beta},\, \sigma^2 (X^\top X)^{-1}). \tag{2.3.11}$$

As a result, we obtain

$$\frac{(n-p)s^2}{\sigma^2} = \frac{\sum_{j=1}^{n} (Y_j - \hat{Y}_j)^2}{\sigma^2} \sim \chi^2_{n-p}. \tag{2.3.12}$$

2.3.2 Hypothesis testing

After we identify the estimates and estimators of the model parameters, we want to check if a particular predictor covariate has a significant influence on the model. Thus, our testing problem is given by

$$H_0 : \beta_j = 0 \text{ and } H_1 : \beta_j \neq 0.$$

If H_0 is true, we have from Equation (2.3.11) and Equation (2.3.12) that the random test statistics

$$T_j = \frac{\hat{\beta}_j}{\hat{\sigma}\sqrt{\nu_j}} \overset{H_0}{\sim} t_{n-p}, \tag{2.3.13}$$

where ν_j is the jth diagonal element of $(X^\top X)^{-1}$. Moreover, it allows us to define a statistical hypothesis test as follows.

Definition 2.3.1. *(Wald tests)*
Suppose we have the linear regression model as given in Equation (2.3.2) and let t_j be the observed of value T_j. Then, testing the significance of one particular coefficient can be done by ***Wald tests*** *as follows*

$$\textit{Reject } H_0 : \beta_j = 0 \textit{ vs } H_1 : \beta_j \neq 0 \textit{ at level } \alpha \iff |t_j| > t_{n-p,1-\frac{\alpha}{2}}, \tag{2.3.14}$$

where $t_{n-p,1-\frac{\alpha}{2}}$ denotes the $1-\frac{\alpha}{2}$ quantile of the univariate Student's t-distribution with $n-p$ degrees of freedom.

Therefore, a large (absolute) value of t_j results in the rejection of the null hypothesis and we keep the corresponding predictor in the model. Otherwise, we remove it from the model.

2.3.3 Subset selection

When we have the response and predictors in the linear regression model, we show how to find the model parameters and test their significance. Nonetheless, we can have a large number of variables as potential predictors and need to select some of them. Subset selection provides us to retain only a subset of the variables, and eliminate the rest from the model. Although there are various strategies for choosing the subset, we benefit from **backward-stepwise selection**.

We start with modeling all potential predictors, so-called **full model**, and sequentially delete the predictor that has the least impact on the fit, which is called as backward-stepwise selection. We drop the covariate with the smallest t_j score given in Equation (2.3.13) as long as the removed covariate is not significant. i.e. we eliminate nonsignificant effects according to the Wald test. Since some covariates are eliminated from the full model, it is called **reduced model**. Nevertheless, we note that the approach does not properly take the multiple testing problems into consideration as stated by Hastie et al. (2009).

2.3.4 Diagnostics and model evaluation

After model fitting, we need to assess whether the assumptions of the linear regression model are fulfilled. Thus, we need some definitions beforehand.

Definition 2.3.2. *(Raw residuals)*
Let y_j denote the observed values of the response and $\hat{y}_j$ denote the model predictions. Then, ***raw residuals*** *are defined as*

$$r_j := y_j - \hat{y}_j \quad for \quad j = 1, ..., n. \tag{2.3.15}$$

Definition 2.3.3. *(Standardized residuals (internally))*
Let y_j denote the observed values of the response and $\hat{y}_j$ denote the model predictions. Then, ***standardized residuals (internally)*** *are defined as*

$$s_j := \frac{r_j}{s\sqrt{1 - h_{jj}}} \quad for \quad j = 1, ..., n, \tag{2.3.16}$$

where s is given in Equation (2.3.10), r_j in Equation (2.3.15) and h_{jj} is jth diagonal element of $X^\top (X^\top X)^{-1} X$.

With the help of these definitions, we can analyze the residuals using visualization and see if they support the assumptions of linearity, independence, normality and constant variances.

The residuals plots we can look at are the following:

1. Residuals versus the fitted values
   It is a scatter plot of the raw residuals on the y axis and *fitted values* (estimated responses) on the x axis. The raw residuals should be randomly scattered around the zero line and they should form a horizontal band thereby satisfying linearity and constant variance assumptions.

2. Normal probability plot of the residuals
   We plot the ordered raw residuals against the ordered quantiles from the standard normal distribution and it should look like a straight line. Isolated points represent unusual observations, whereas a curved line demonstrates that the raw residuals are probably not normally distributed.

3. Residuals versus predictor values
   It is a scatter plot of the raw residuals on the y axis and the predictor values on the x axis. The well-behaved plot will bounce randomly and form a roughly horizontal band around the zero line. Otherwise, we should transform the respective predictor and fit the model again.

4. Standardized residuals versus fitted values, Scale-Location
   It is a scatter plot of the standardized residuals on the y axis and *fitted values* on

the x axis. We check if the standardized residuals suffer from non-constant variance, i.e. **heteroscedasticity**. The more horizontal the line in the graph is, the more likely the data is **homoscedastic**, i.e. the standardized residuals have the equal variance.

2.4 Harmonic regression

After discussing the linear regression model, we explain another model, called harmonic regression model in this part. Many time series show seasonal or cyclical effects of assorted kinds. In order to investigate such periodic phenomena, as well as performing functions like forecasting, we need to estimate the periodic components and model them. We can accomplish our aims by first generating the harmonics using suitable sine and cosine transformations and then including them as the covariates in the linear regression model as discussed in Section 2.3. Generating the appropriate sine and cosine terms and modeling them are called **harmonic regression**, which is discussed in Chatfield et al. (1988) and Bloomfield (2000). Furthermore, more details about the harmonic regression model and parameter estimation can be found in Arti et al. (2007).

Let X_t for $t = 1, ..., T$ denote a time series, e.g. medication sales on consecutive weeks. Suppose X_t has a periodic component, then we can formulate it as

$$X_t = \mu + \rho \cos(\omega t - \theta) + \epsilon_t \quad \text{for} \quad t = 1, ..., T, \tag{2.4.1}$$

where

- μ is the mean of the series,
- ρ is the amplitude,
- ω is the angular velocity or frequency,
- t is time,
- θ is the phase displacement,
- ϵ_t is the random error of the series about the harmonic component

The frequency is measured in radians per unit period and the quantity $2\pi/\omega$ measures the period. The phase displacement, which is similarly measured in radians, exhibits the extent to which the cosine function has been displaced by a shift along the time axis. By using the compound-angle formula,

$$\cos(a - b) = \cos(a)\cos(b) + \sin(a)\sin(b), \tag{2.4.2}$$

we can rewrite the periodic component equation as

$$X_t = \mu + \rho \cos\theta \cos(\omega t) + \rho \sin\theta \sin(\omega t) + \epsilon_t \tag{2.4.3}$$

$$= \mu + \alpha \cos(\omega t) + \beta \sin(\omega t) + \epsilon_t, \tag{2.4.4}$$

where

$$\alpha = \rho \cos\theta,$$
$$\beta = \rho \sin\theta,$$
$$\rho^2 = \alpha^2 + \beta^2.$$

Moreover, X_t can consists of multiple periodic components and we can model the seasonal fluctuation comprising the full set of harmonically related frequencies as

$$X_t = \mu + \sum_{j=0}^{p} [\alpha_j \cos(\omega_j t) + \beta_j \sin(\omega_j t)] + \epsilon_t \quad \text{for} \quad t = 1, ..., T. \tag{2.4.5}$$

On the other hand, if $p = 0$ is taken, Equation (2.4.5) consists of three unknown parameters to be estimated, which are α, β and ω.

2.4.1 Parameter estimation: ω

First of all, we need to determine the appropriate set of frequencies to be fitted in the harmonic regression model. One idea could be to investigate the effect of precise periods. For instance, suppose we have a weekly data of three years. Then the periods such as $T_1 = 52$, $T_2 = 12$ and $T_3 = 4$ would account for the impact of year, month and quarter, respectively. Nevertheless, the data may not exhibit such seasonal patterns and we cannot capture the correct harmonic terms via this approach. However, we investigate the effect of these terms in Section 4.1. Alternatively we can benefit from **spectral analysis** and inspect **periodogram** to determine the appropriate frequencies, ω_j's. Here we refer to Koopmans (2004) for more details.

We need some tools to explain spectral analysis in detail and the Fourier decomposition of a series, which explains the time series entirely as a composition of sinusoidal functions, is the first tool we introduce. It allows us to represent an element of the time series as

$$X_t = \sum_{j=0}^{p} [\alpha_j \cos(\omega_j t) + \beta_j \sin(\omega_j t)] \quad \text{for} \quad t = 1, ..., T. \tag{2.4.6}$$

If we assume that $T = 2p$ is even, this sum leads to T functions whose frequencies

$$\omega_j = \frac{2\pi j}{T} \quad \text{for} \quad j = 0, ..., p = \frac{T}{2} \tag{2.4.7}$$

are at equally spaced points in the range $[0, \pi]$. Furthermore, the angular velocity ω_j relates to a pair of trigonometrical terms achieving i cycles in the T periods spanned by the data set.

To calculate the Fourier coefficients, α_j and β_j in Equation (2.4.6), we apply the linear regression estimation approach, introduced in Section 2.3, to Equation (2.4.6). Note that there is not any regression residual in this case, because we estimate T coefficients from T data points.

Hence, let $\boldsymbol{a_j} = [a_{1j}, ..., a_{(T)j}]^\top$ and $\boldsymbol{b_j} = [b_{1j}, ..., b_{(T)j}]^\top$ be the vector of T values of the functions $\cos(\omega_j t)$ and $\sin(\omega_j t)$ for $j = 0, ..., p = \frac{T}{2}$, respectively. Then the orthogonality

conditions of $\cos(\omega_j t)$ and $\sin(\omega_j t)$ induce

$$\boldsymbol{a_k}^\top \boldsymbol{a_j} = 0, \quad \text{if} \quad k \neq j, \quad \text{for} \quad j,k = 0,...,p = \frac{T}{2}, \tag{2.4.8}$$

$$\boldsymbol{b_k}^\top \boldsymbol{b_j} = 0, \quad \text{if} \quad k \neq j, \quad \text{for} \quad j,k = 0,...,p = \frac{T}{2}, \tag{2.4.9}$$

$$\boldsymbol{a_k}^\top \boldsymbol{b_j} = 0, \quad \text{for all} \quad k,j \quad \text{for} \quad j,k = 0,...,p = \frac{T}{2}. \tag{2.4.10}$$

Moreover, the following equalities hold

$$\boldsymbol{a_0}^\top \boldsymbol{a_0} = \boldsymbol{a_p}^\top \boldsymbol{a_p} = T, \tag{2.4.11}$$

$$\boldsymbol{b_0}^\top \boldsymbol{b_0} = \boldsymbol{b_p}^\top \boldsymbol{b_p} = 0, \tag{2.4.12}$$

$$\boldsymbol{a_j}^\top \boldsymbol{a_j} = \boldsymbol{b_j}^\top \boldsymbol{b_j} = \frac{T}{2}, \quad \text{for} \quad j = 1,...,p-1. \tag{2.4.13}$$

If we denote the observed values of the time series X_t by $\boldsymbol{x} = (x_1,...,x_T)^\top$ denote, then the Fourier coefficients, in Equation (2.4.6), are estimated by the linear regression as follows:

$$\alpha_0 = (\boldsymbol{a_0}^\top \boldsymbol{a_0})^{-1} \boldsymbol{a_0}^\top \boldsymbol{x} = \frac{1}{T}\sum_{t=1}^{T} x_t = \overline{x}, \tag{2.4.14}$$

$$\beta_0 = 0, \tag{2.4.15}$$

$$\alpha_j = (\boldsymbol{a_j}^\top \boldsymbol{a_j})^{-1} \boldsymbol{a_j}^\top \boldsymbol{x} = \frac{2}{T}\sum_{t=1}^{T} x_t \cos(\omega_j t) \quad \text{for} \quad j = 1,...,p, \tag{2.4.16}$$

$$\beta_j = (\boldsymbol{b_j}^\top \boldsymbol{b_j})^{-1} \boldsymbol{b_j}^\top \boldsymbol{x} = \frac{2}{T}\sum_{t=1}^{T} x_t \sin(\omega_j t) \quad \text{for} \quad j = 1,...,p. \tag{2.4.17}$$

Besides, Equation (2.4.6) can be formulated as

$$\boldsymbol{x}^\top \boldsymbol{x} = \alpha_0^2 + \sum_{j=1}^{p} \alpha_j^2 \boldsymbol{a_j}^\top \boldsymbol{a_j} + \sum_{j=1}^{p} \beta_j^2 \boldsymbol{b_j}^\top \boldsymbol{b_j}. \tag{2.4.18}$$

From Equation (2.4.18) it follows

$$(\boldsymbol{x} - \overline{\boldsymbol{x}})^\top (\boldsymbol{x} - \overline{\boldsymbol{x}}) = \frac{T}{2}\sum_{j=1}^{p} \{\alpha_j^2 + \beta_j^2\}. \tag{2.4.19}$$

Thus, we can express the sample variance by

$$\frac{1}{T}\sum_{t=1}^{T} (x_t - \overline{x})^2 = \frac{1}{2}\sum_{j=1}^{p} \{\alpha_j^2 + \beta_j^2\}. \tag{2.4.20}$$

In particular

$$\sum_{t=1}^{T} (\boldsymbol{x}_t - \overline{\boldsymbol{x}})^2 = \sum_{j=1}^{p} \frac{T}{2}\{\alpha_j^2 + \beta_j^2\} =: \sum_{j=1}^{p} f(\omega_j). \tag{2.4.21}$$

Therefore the term $(\alpha_j^2 + \beta_j^2)/2$ corresponds to the proportion of the variance, which is attributable to the trigonometric terms at frequency ω_j, and it allows us to define the periodogram as follows:

Definition 2.4.1. *(Periodogram)*
Let X_t denote a time series for $t = 1, ..., T$ satisfying Equations (2.4.6) and (2.4.7). Then, the graph of the function

$$f(\omega_j) := \frac{T}{2}(\alpha_j^2 + \beta_j^2) \tag{2.4.22}$$

is called as ***periodogram****.*

Even though the periodogram displays the power of frequencies within the data set, when the data does not complete an integer number of a periodic component within itself, the periodogram fails to show the correct frequencies, which is so-called spectral leakage problem as shown in 2.4.1. Therefore, we need a technique preventing the spectral leakage problem; however, we need some definitions before introducing one of the known techniques.

Example 2.4.1. *Suppose we have $n_1 = 50$ and $n_2 = 59$ observations generated from the process $X_t = sin(2\pi\omega t)$, where $\omega = 0.04$. Then, the peak in the periodogram should have occurred at $\omega = 0.04$. However, as can be seen in Figure 2.4.1, second sample fails to show $\omega = 0.04$ owing to the fractional number of the periodic component, which is spectral leakage problem.*

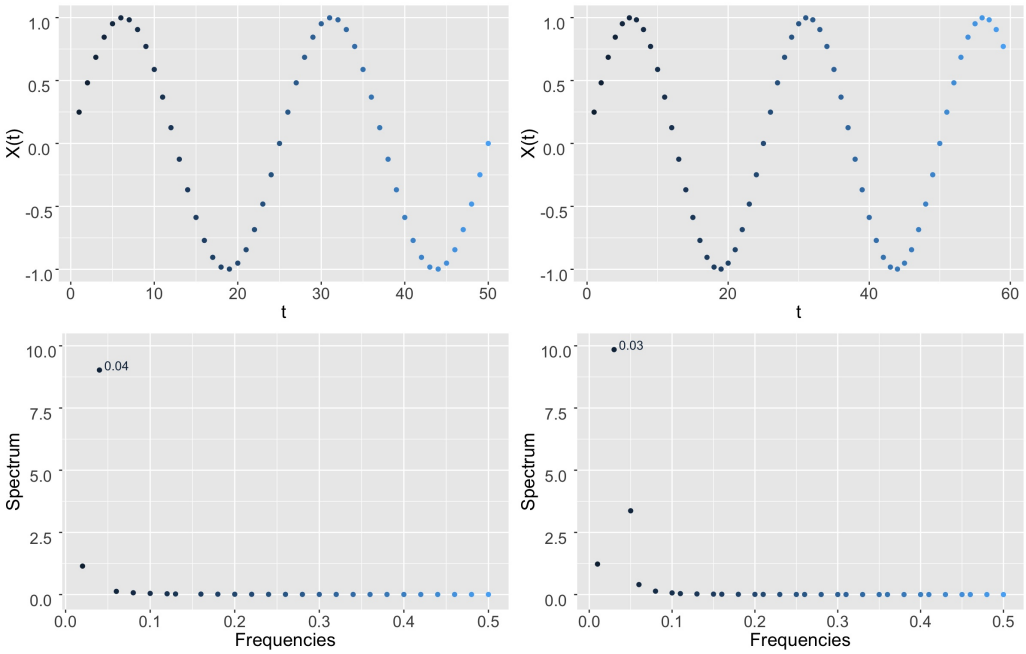


Figure 2.4.1: *Left/right*: *First/second* sample and their periodograms displaying the frequency value at the maximum spectrum. Right sample represents spectral leakage problem.

Definition 2.4.2. *(AR(p) model)*
Let $(X_t)_{t\in\mathbb{N}}$ denote a time series. Then, the ***AR(p) model*** *is defined as*

$$X_t = c + \sum_{j=1}^{p} a_j X_{t-j} + \epsilon_t, \tag{2.4.23}$$

where c is a constant, $a_1, ..., a_p$ are the AR-parameters with $a_p \neq 0$ and error term ϵ_t is a series of uncorrelated random variables with $E(\epsilon) = 0$ and finite variance $E(\epsilon^2) = \sigma^2 < \infty$.

Definition 2.4.3. *(Autocovariance function)*
Let $(S_t)_{t\in\mathbb{N}}$ denote a stochastic process. Then, the ***autocovariance*** *of the process is defined as*

$$\gamma(h) := Cov(S_{n+h}, S_h), \tag{2.4.24}$$

where $Cov(S_1, S_2)$ is the ***covariance*** *between S_1 and S_2 given by*

$$Cov(S_1, S_2)_{=}\mathbb{E}(S_1 S_2) - \mathbb{E}(S_1)\mathbb{E}(S_2). \tag{2.4.25}$$

Definition 2.4.4. *(Spectral density)*
Let $(X_t)_{t\in\mathbb{N}}$ denote a time series. If its autocovariance γ satisfies $\sum_{h=-\infty}^{\infty} |\gamma(h)| < \infty$, then its ***spectral density*** *is defined as*

$$f(x) := \sum_{h=-\infty}^{\infty} \gamma(h) \exp\{-2\pi i x h\}, \tag{2.4.26}$$

where $-\infty < x < \infty$, i is the imaginary unit defined by the property $i^2 = -1$.

To avoid spectral leakage problem, we can calculate **spectral densities via autoregressive process of order p** as explained by Venables et al. (2002).

Definition 2.4.5. *(Spectral density of an AR(p) process)*
Suppose we have the parameters as characterized in Equation (2.4.23). Then, the ***spectral density of the AR(p) process*** *is given by*

$$s(\omega) := \frac{\sigma^2}{|1 - \sum_{j=1}^{p} a_j \exp\{-ij\omega\}|^2}\ , \tag{2.4.27}$$

where i is the imaginary unit defined by the property $i^2 = -1$.

Moreover, because $\exp\{ix\} = \cos(x) + i\sin(x)$ and $|a + ib| = \sqrt{a^2 + b^2}$ hold for the imaginary unit, i, Equation (2.4.27) can be rewritten as:

$$s(\omega) = \frac{\sigma^2}{|1 - \sum_{j=1}^{p} a_j(\cos(\omega j) + i\sin(\omega j))|^2}\ . \tag{2.4.28}$$

And it follows that

$$s(\omega) = \frac{\sigma^2}{\left(1 - \sum_{j=1}^{p} a_j \cos(\omega j)\right)^2 + \left(\sum_{j=1}^{p} a_j \sin(\omega j)\right)^2}\ . \tag{2.4.29}$$

Example 2.4.2. *Suppose we have the same* $n_1 = 50$ *and* $n_2 = 59$ *observations as given in Figure 2.4.1. We fit the best AR(p) model to them first and then analyze their spectral densities. As can be seen, spectral leakage problem is solved since both samples have their maximum spectrum value at* $\omega = 0.04$.

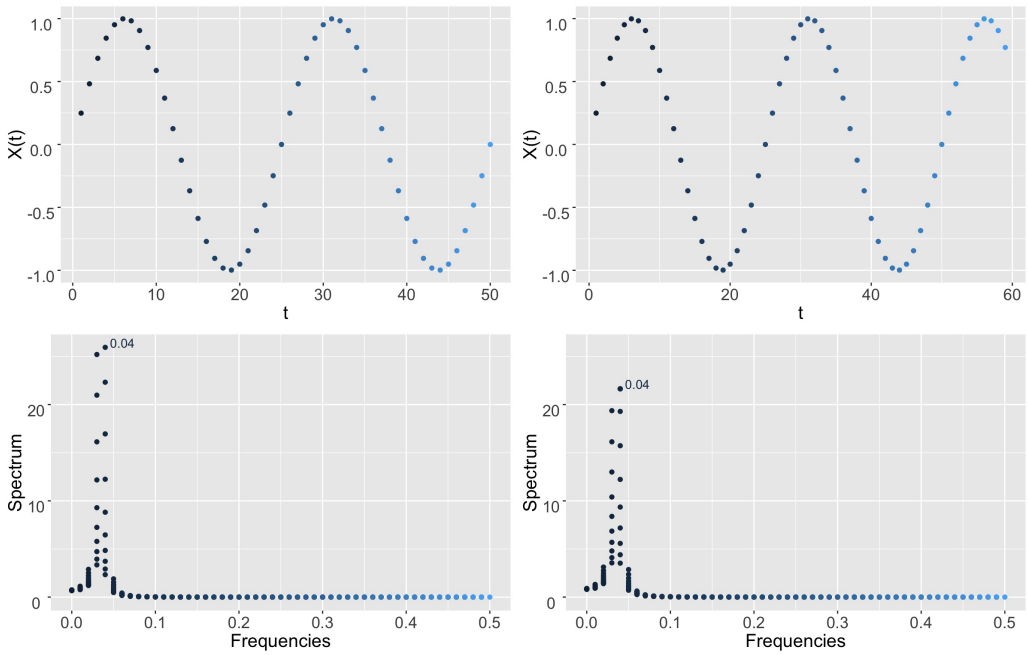


Figure 2.4.2: *Left/right*: *First/second* sample and their AR(p) processed spectral densities displaying the frequency value at the maximum density. The p values in these samples are 2 and 4, respectively. Spectral leakage problem is handled by AR(p) model when compared to Figure 2.4.1.

For more details about AR models we refer to Brockwell et al. (2016). All in all, we fit the best AR(p) model to the data, where $p \in [1, 5]$ is assumed and the best model is chosen by the AIC score, defined in Equation (2.5.4), and afterwards we compute the spectral density of the fitted model. We compare the results with periodogram and choose either of them. Finally, we take the four frequencies, whose densities are higher than the others, and evaluate their effects via the harmonic regression model. We determine the number of frequencies to be included in the harmonic regression model as four by trial and error. Our model now reads

$$X_t = \mu + \sum_{j=0}^{3} [\alpha_j \cos(\omega_j t) + \beta_j \sin(\omega_j t)] + \epsilon_t \quad \text{for} \quad t = 1, ..., T \tag{2.4.30}$$

with parameters stated in Equation (2.4.1).

2.4.2 Parameter estimation: α and β

After we determined the appropriate frequency $\boldsymbol{\omega}$ in Equation (2.4.30), we are left with finding the remaining parameters, $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$.

For this goal we use the linear regression model. Durbin (1960) has justified the usage of linear regression estimates of $\boldsymbol{\alpha} = (\alpha_0, \alpha_1, \alpha_2, \alpha_3)^\top$ and $\boldsymbol{\beta} = (\beta_0, \beta_1, \beta_2, \beta_3)^\top$ for the harmonic regression model of the type in Equation (2.4.30).

2.5 Performance measures for comparing statistical models

After discussing the models we use in the thesis, here we define some performance measures which we later use in the comparison of the different models. Moreover, we distinguish between in sample and out of sample performance measures. In the thesis, in sample corresponds to the training set, whereas out of sample corresponds to the test set.

In sample performance measures

Definition 2.5.1. *(R^2, R^2_{adj})*
Suppose we have a linear regression model and the total number of estimated regression parameters is p. Let $\boldsymbol{y} = (y_1, ..., y_n)^\top \in \mathbb{R}^n$ *be a vector of the observed values of the response and* $\hat{\boldsymbol{y}} = (\hat{y}_1, ..., \hat{y}_n)^\top \in \mathbb{R}^n$ *the vector of the corresponding predictions. Besides, let*

$$\overline{y} := \frac{\sum_{j=1}^{n} y_j}{n} \tag{2.5.1}$$

denote the ***mean*** *of the observations. Then the* ***multiple coefficient of determination*** R^2 *is defined as*

$$\mathrm{R}^2 := \frac{\sum_{j=1}^{n}(\hat{y}_j - \overline{y})^2}{\sum_{j=1}^{n}(y_j - \overline{y})^2}. \tag{2.5.2}$$

In addition, we define the ***adjusted multiple coefficient of determination*** R^2_{adj} *as*

$$\mathrm{R}^2_{\mathrm{adj}} := 1 - \frac{\sum_{j=1}^{n}(y_j - \hat{y}_j)^2 / (n-p)}{\sum_{j=1}^{n}(y_j - \overline{y})^2 / (n-1)}. \tag{2.5.3}$$

R^2 represents the proportion of the variance in the response which is explained by the regression model and it takes values between zero and one. The closer it is to one, the better our model accounts for the variability in the response. Nonetheless, adding predictor variables to the linear models always increases R^2; thus, R^2_{adj} is preferred to compare the goodness of fit for models since it takes the number of estimated regression parameters into account. The use of R^2_{adj} was justified by Theil (1961).

Definition 2.5.2. *(AIC)*
Suppose that we have a statistical model of some data and let k be the number of estimated parameters in the model. Moreover, let $\hat{L}$ *be the maximum value of the likelihood function for the model. The* ***Akaike information criterion (AIC)*** *for the model is given by*

$$\mathrm{AIC} := 2k - 2\ln(\hat{L}). \tag{2.5.4}$$

AIC was first developed by Akaike (1973) as a way to compare different nested models on a given outcome. We call models $M_1 \subset M_2 \subset ... \subset M_K$ nested if model M_j is a sub-model of M_{j+1} for $j = 1, ..., K-1$. For a set of nested models, we choose the model with the lowest AIC score.

Out of sample performance measures

Definition 2.5.3. *(MSPE, RMSPE)*
Let $\boldsymbol{y} = (y_1, ..., y_n)^\top \in \mathbb{R}^n$ *be a vector of the observed values of the variable being predicted and* $\hat{\boldsymbol{y}} = (\hat{y}_1, ..., \hat{y}_n)^\top \in \mathbb{R}^n$ *the vector of corresponding predictions. Then the* ***mean squared prediction error (MSPE)*** *computed on* q *data points, which were not utilized in the model fitting with* $q < n$*, is given by*

$$\text{MSPE} := \frac{1}{q} \sum_{j=n-q+1}^{n} (y_j - \hat{y}_j)^2. \tag{2.5.5}$$

Furthermore, we define the ***root mean squared prediction error (RMSPE)*** *as*

$$\text{RMSPE} := \sqrt{\text{MSPE}}. \tag{2.5.6}$$

Even though MSPE and RMSPE are common measures of the uncertainty in forecasting, they can be problematic since they are highly sensitive to outliers. For instance, only 5 out of the 1001 series of the M-Competition resolved the value of RMSPE due to the extreme errors, on the other hand the remaining 996 had much less impact as shown by Chatfield et al. (1988).

2.6 Clustering

Clustering is one of the most common methods in unsupervised learning, where every data point is classified without predetermining its labels. It aims to identify hidden structures or similar groups in data. There are many practical applications of clustering like in the fields of marketing and pattern recognition. It helps, for example, companies to have customer segmentation.

To perform clustering, one must usually decide on four different components: a distance measure to quantify if the data points in the data are similar, a method to obtain clusters, features of data points to calculate the distance or to model the data and the desired number of clusters to return a clustering. Depending on the cluster algorithm that is performed, the execution order of the components can change. For instance, some methods first of all predetermine the features of the cluster analysis, whereas other methods decide on them in conjunction with the desired number of clusters as discussed in Dy et al. (2004). Furthermore, some methods such as model-based approaches do not use a distance measure.

Most clustering analyses, on the other hand, perform on static data, where all feature values do not change with time or change negligibly. Therefore, when obtaining a clustering with time series data that is not static, the existing algorithms for clustering static data are changed in such a way that the time series data is converted into the form of static data so that the existing algorithms for clustering static data can be directly applied. This time series clustering approach is known as feature-based approach. The other approaches, raw-data-based and model-based approach, as well as the recent studies about the clustering of the time series data can be found in the survey by Warren Liao (2005).

When performing a feature-based approach in the clustering of the time series data, some features from the data are selected. For instance, Nanopoulos et al. (2001) used the mean, standard deviation, skewness and kurtosis of the time series for the classification purpose, whereas Wang et al. (2007) introduced some features that consist of measures of trend, seasonality, periodicity, serial correlation, skewness, kurtosis, chaos, nonlinearity, and self-similarity to represent the time series. Nevertheless, there is not a unique set of features that performs well in all clustering analyses, i.e. the good feature selection is application dependent.

Since our aim is to obtain a clustering with the time series data in the thesis, we follow feature-based approach we explained above and use the mean and standard deviation of the time series data as the features. Then we calculate the Euclidean distance given in Equation (2.6.1) between the selected features. The justification and more details can be found in Section 6. After deciding on the features and the distance measure, we address the distance-based clustering approach and discuss several algorithms for obtaining a clustering. When performing these algorithms, we need to determine the 'optimal' number of clusters. Thus, we discuss several indices for this goal and compare them with each other.

For more information about the cluster analysis, we refer to the book by Everitt et al. (2011).

Definition 2.6.1. *(Euclidean distance)*
Let $\boldsymbol{x} = (x_1, ..., x_d)^\top \in \mathbb{R}^n$ *and* $\boldsymbol{y} = (y_1, ..., y_d)^\top \in \mathbb{R}^n$ *be a d-dimensional vector of the continuous observations. Then the Euclidean distance between* $\boldsymbol{x}$ *and* $\boldsymbol{y}$ *is given by:*

$$d_{euc}(\boldsymbol{x}, \boldsymbol{y}) := \sqrt{\sum_{j=1}^{d} (x_j - y_j)^2}. \tag{2.6.1}$$

2.6.1 Overview of clustering methods

In this section, we give an overview of several clustering methods. Although there are many clustering algorithms and their categorization in the literature, we follow the classification of the clustering methods given by Han et al. (2011): partitional clustering, hierarchical clustering, density-based clustering, grid-based clustering and model-based clustering. Because all methods except the model-based clustering mainly use distance criteria to perform a clustering, they can be grouped into **distance-based clustering** methods.

Additionally, clustering methods can be divided into three groups according to their clustering assignments. For instance, assigning each observation to a single cluster is known as **exclusive clustering**, while if each observation belongs to every clustering with a probability is called **fuzzy clustering**. Thus, assigning each observation to the cluster, in which the assignment probability is highest, converts the fuzzy clustering into the exclusive clustering. On the other hand, **overlapping** or **nonexclusive clustering** methods allow an observation to belong to any of clusters.

Even though there are many clustering methods and their classifications, we recall that the main requirements of clustering methods are scalability and the ability to deal with different types of attributes and noisy data. Furthermore, interpretability and usability are crucial.

In our analysis, we have six real valued, continuous features as can be seen in Section 6.1. Therefore, we focus on the interpretability and usability performance of the clustering methods. Because the partitional and hierarchical clustering appear to be most commonly used clustering methods in the time series clustering as Warren Liao (2005) concluded, we perform these two types of the distance-based methods. Moreover, we compare their performances.

Before we discuss each method in the upcoming sections, we give an overview of the methods that we will use later in Table 2.6.1.

Method	General properties
Partitioning clustering	Divide data into groups such that each group must contain at least one observation. Distance-based clustering. Find exclusive clusters of spherical shape. Due to exhaustive e-numeration of partition, many popular heuristics such as k-means exist.
Hierarchical clustering	Create hierarchal decomposition of the data. Usually distance-based clustering. Find exclusive clusters. Cannot correct erroneous merges.

Table 2.6.1: Overview of the clustering methods that we applied in the thesis.

2.6.2 Distance-based clustering methods

We start with explaining distance-based clustering methods. As we mentioned earlier, these methods use distance criteria to obtain a clustering and several different techniques such as partitional and hierarchical clustering can be put into this category. What we see as a standard goal in all these methods, is that given an underlying set of points, partition them into a collection of homogenous groups so that points in the same group are close together, whereas points in different clusters are far apart. On the other hand, we can observe that the underlying principles of them are quite diverse and they often result in qualitatively different results as we showed in Section 6.2.

In the upcoming subsections we discuss the underlying principles of partitional and hierarchical clustering methods that we utilize in thesis. However, we present first an **axiomatic framework** for clustering, where one enumerates a collection of simple properties that a solution should satisfy and then analyzes how these characteristics constrain the solution one can obtain. The framework is developed by Kleinberg (2002) to indicate a set of basic trade-offs that are inherent in the clustering problem and to distinguish between clustering methods. The reason why we introduce it is that there has been relatively small number of research that aim to question clustering independently of any specific method, objective function or generative model and the idea of no universal clustering algorithm is partially captured by this framework, e.g. **impossibility theorem**. Next, we give definitions that we need for the impossibility theorem and finally the theorem itself.

Definition 2.6.2. *(Clustering algorithm)*
*A **clustering algorithm** is a function $\mathfrak{C}$ that converts a metric space (X, d_X) into a partition $\mathcal{C}_X$ of its points into 'clusters'.*

Theorem 2.6.1. *(Kleinberg's impossibility theorem)*
For each $|X| \geq 2$, there exists no clustering function $\mathfrak{C}$ that satisfies:

1. *Scale-invariance: $\mathfrak{C}(X, d_X) = \mathfrak{C}(X, \lambda \cdot d_X)$ for any metric d_X on X and $\lambda > 0$.*
2. *Richness: For any partition $\mathcal{C}_X$ of X there is some metric d_X on X so that $\mathfrak{C}(X, d_X) = \mathcal{C}_X$.*
3. *Consistency: Let d'_X be another metric on X such that it holds for all $j, k \in X$*

belonging to the same cluster of $\mathcal{C}_X$, we have $d'_X(j,k) \leq d_X(j,k)$ and for all $j,k \in X$ belonging to different clusters of $\mathcal{C}_X$, we have $d'_X(j,k) \geq d_X(j,k)$. Next, if $\mathfrak{C}(X, d_X) = \mathcal{C}_X$, then $\mathfrak{C}(X, d'_X) = \mathcal{C}_X$.

Proof The proof can be found in the paper by Kleinberg (2002).

In other words, scale-invariance is the simple requirement that the clustering function should not be sensitive to changes in the units of distance measurement. A second property, richness, is that every partition of X must be a possible output. Finally, consistency implies that cluster stays the same after reducing the distance within cluster and enlarging distance between cluster. With these three axioms, what Kleinberg's impossibility theorem presents, is that there does not exist any clustering algorithm that satisfies scale-invariance, richness and consistency. Thus, there are trade-offs that we encounter when applying different methods. For instance, k-means algorithm that we explain below does not meet the requirement of consistency property.

Partitional clustering: *k*-means

The most fundamental version of distance-based clustering is partitioning, which organizes data into several exclusive clusters. Although there are many different methods within partitional clustering, we apply k-means technique due to its usability and interpretability.

Suppose a data set X contains n observations in Euclidean space, i.e. $X = \{\boldsymbol{x}_1, ..., \boldsymbol{x}_n\} \subset \mathbb{R}^d$ and k-means method distributes the observations in X into k clusters. We denote the clusters by $\mathcal{C} = \{\mathcal{C}_1, ..., \mathcal{C}_k\}$ and the cluster centroids by $\boldsymbol{\mu} = \{\boldsymbol{\mu}_1, ..., \boldsymbol{\mu}_k\} \subset \mathbb{R}^d$. The centroid is the mean of the data points assigned to the cluster. Additionally, suppose it holds that $\mathcal{C}_j \subset X$ and $\mathcal{C}_j \cap \mathcal{C}_l = \varnothing$ for $1 \leq j, l \leq k$ and $j \neq l$. Then the quality of cluster $\mathcal{C}_j$ can be measured by the **within-cluster variation**, which is the squared Euclidean distance between the all observations in $\mathcal{C}_j$ and the cluster centroid $\boldsymbol{\mu}_j$, given in Equation (2.6.1). Thus, the ultimate goal of k-means is to minimize the 'total' within-cluster variance that we define in Equation (2.6.2).

Definition 2.6.3. *(Total within-cluster variance)*
*Let $X = \{\boldsymbol{x}_1, ..., \boldsymbol{x}_n\} \subset \mathbb{R}^d$ denote a set of observations. Suppose the clustering method partitions them into k clusters, $\mathcal{C} = \{\mathcal{C}_1, ..., \mathcal{C}_k\}$. Let denote additionally the cluster centroids by $\boldsymbol{\mu} = \{\boldsymbol{\mu}_1, ..., \boldsymbol{\mu}_k\} \subset \mathbb{R}^d$, where $\boldsymbol{\mu}_j$ is the centroid of the cluster $\mathcal{C}_j$ for $1 \leq j \leq k$. Then **the total within cluster variance**, E, is given by*

$$E\left(\mathcal{C}_1, ..., \mathcal{C}_k\right) := \sum_{j=1}^{k} \sum_{\boldsymbol{x}_j \in \mathcal{C}_j} [d_{euc}(\boldsymbol{x}_j, \boldsymbol{\mu}_j)]^2. \tag{2.6.2}$$

k-means is an iterative method. For each cluster, it computes the new centroids using the observations that were assigned to the cluster in the previous iteration. All the observations are then reassigned using the updated centroids as the new cluster centroids. The iterations continue until the assignment is stable, that is, the clusters formed in the current step are the same as those formed in the previous step. However, obtaining the

best clustering with k-means is NP-hard even in the 2-D Euclidean space as shown by Mahajan et al. (2009) and the final clustering is sensitive to initialization of the centroids. Hence, many greedy algorithms such as Lloyd, MacQueen and Hartigan-Wong exist. In the thesis we follow the Hartigan-Wong algorithm introduced by Hartigan et al. (1979) because Telgarsky et al. (2010) reported that Hartigan-Wong algorithm is likely to find more refined clusters. We give the steps in Algorithm 1.

Algorithm 1 k-means with Hartigan-Wong algorithm

Input: A set of data observations $X = \{\boldsymbol{x}_1, ..., \boldsymbol{x}_n\} \subset \mathbb{R}^d$, a distance function $d : (X, X) \to \mathbb{R}$, a random clustering partition $\mathcal{C} = \{\mathcal{C}_1, ..., \mathcal{C}_k\}$, a maximum number of iterations m_{max} and an iteration indicator $m = 0$.
Output: The clustering partition $\mathcal{C} = \{\mathcal{C}_1, ..., \mathcal{C}_k\}$ that minimizes the total within-cluster variance defined in Equation (2.6.2).
for $j = 1, ..., k$ **do**
    $\boldsymbol{\mu}_j = \frac{1}{|\mathcal{C}_j|} \sum_{\boldsymbol{x}_j \in \mathcal{C}_j} \boldsymbol{x}_j$
end for
Select a random permutation of $\{1, ..., n\}$ given by $\pi := \{\pi(1), ..., \pi(n)\}$
while not Done **do**
    Done = TRUE
    for i = 1, ..., n **do**
        m = m +1
        Determine cluster $\mathcal{C}^*$ to which $\boldsymbol{x}_{\pi(i)}$ belongs
        $\mathcal{C}^*_- := \mathcal{C}^* \setminus \left\{\boldsymbol{x}_{\pi(i)}\right\} =: \mathcal{C}_{j(\boldsymbol{x}_{\pi(i)})}$
        $\boldsymbol{\mu}^*_- = \frac{1}{|\mathcal{C}^*_-|} \sum_{\boldsymbol{x}_l \in \mathcal{C}^*_-} \boldsymbol{x}_l$
        $\mathrm{E}_1(\mathcal{C}_1, ..., \mathcal{C}_k) := \mathrm{E}(\mathcal{C}_1 \cup \left\{\boldsymbol{x}_{\pi(i)}\right\}, \mathcal{C}_2, ..., \mathcal{C}_{j(\boldsymbol{x}_{\pi(i)})-1}, \mathcal{C}^*_-, \mathcal{C}_{j(\boldsymbol{x}_{\pi(i)})+1}, ..., \mathcal{C}_k)$
        $\mathrm{E}_2(\mathcal{C}_1, ..., \mathcal{C}_k) := \mathrm{E}(\mathcal{C}_1, \mathcal{C}_2 \cup \left\{\boldsymbol{x}_{\pi(i)}\right\}, ..., \mathcal{C}_{j(\boldsymbol{x}_{\pi(i)})-1}, \mathcal{C}^*_-, \mathcal{C}_{j(\boldsymbol{x}_{\pi(i)})+1}, ..., \mathcal{C}_k)$
        ...
        $\mathrm{E}_k(\mathcal{C}_1, ..., \mathcal{C}_k) := \mathrm{E}(\mathcal{C}_1, \mathcal{C}_2, ..., \mathcal{C}_{j(\boldsymbol{x}_{\pi(i)})-1}, \mathcal{C}^*_-, \mathcal{C}_{j(\boldsymbol{x}_{\pi(i)})+1}, ..., \mathcal{C}_k \cup \left\{\boldsymbol{x}_{\pi(i)}\right\})$
        $j_{min} = \mathrm{argmin}_{j=1,...,k} \mathrm{E}_j(\mathcal{C}_1, ..., \mathcal{C}_k)$
        if $j_{min} \neq j$ **then**
            Done = FALSE
            Change the clustering: $\mathcal{C}_+ := \mathcal{C}_{j_{min}} \cup \left\{\boldsymbol{x}_{\pi(i)}\right\}$
            $\mathcal{C} = \left\{\mathcal{C}_+, \mathcal{C}^*_-, (\mathcal{C} \setminus \{\mathcal{C}_{j_{min}}, \mathcal{C}^*\})\right\}$
        else if m = m_{max} **then**
            Done = FALSE
        end if
    end for
end while

Hierarchical clustering: agglomerative clustering

While partitioning methods divide the data into exclusive clusters, in some cases we may want to partition our data into groups at different levels such as in a hierarchy. A hierarchical clustering groups the data observations into a hierarchy or 'tree' of clusters and it can be either agglomerative or divisive, depending on whether hierarchy is formed in a bottom-up or top-down manner. In the thesis we apply agglomerative clustering method, bottom-up strategy, and give its steps in Algorithm 2 after we introduce some definitions and concepts.

In the **agglomerative clustering** each data observation forms its own cluster and the clusters are iteratively merged into larger clusters, until all the observations are in a single cluster or certain termination conditions are satisfied. The single cluster becomes the hierarchy's root. For the merging step, the two clusters that are closest to each other (according to some similarity measure that we define in Equations (2.6.3) and (2.6.4)) are determined and merged to form one cluster. As a result, two clusters are merged per iteration in the agglomerative clustering, where each cluster contains at least one observation.

Definition 2.6.4. *(Average linkage)*
Let $X = \{\boldsymbol{x}_1, ..., \boldsymbol{x}_n\} \subset \mathbb{R}^d$ denote a set of observations and $\mathcal{C} = \{\mathcal{C}_1, ..., \mathcal{C}_k\}$ denote its partition into k clusters. Moreover, define a distance function $d : (X, X) \to \mathbb{R}$. Then the average linkage between two clusters $\mathcal{C}_j$ and $\mathcal{C}_i$ $(1 \leq i, j \leq k, i \neq j)$ is given by

$$d_{avg}(\mathcal{C}_j, \mathcal{C}_i) := \frac{1}{|\mathcal{C}_j| \cdot |\mathcal{C}_i|} \sum_{\boldsymbol{x}_j \in \mathcal{C}_j} \sum_{\boldsymbol{x}_i \in \mathcal{C}_i} d(\boldsymbol{x}_j, \boldsymbol{x}_i). \tag{2.6.3}$$

Definition 2.6.5. *(Ward's method)*
Let $X = \{\boldsymbol{x}_1, ..., \boldsymbol{x}_n\} \subset \mathbb{R}^d$ denote a set of observations and $\mathcal{C} = \{\mathcal{C}_1, ..., \mathcal{C}_k\}$ denote its partition into k clusters. Moreover, define a distance function $d : (X, X) \to \mathbb{R}$. Then the Ward's methods defines the distance between two clusters $\mathcal{C}_j$ and $\mathcal{C}_i$ $(1 \leq i, j \leq k, i \neq j)$ as the merging cost of combining two clusters, i.e.

$$d_{ward}(\mathcal{C}_j, \mathcal{C}_i) := \sum_{\boldsymbol{x}_j \in \mathcal{C}_j \cup \mathcal{C}_i} d(\boldsymbol{x}_j, \boldsymbol{\mu}_{\mathcal{C}_j \cup \mathcal{C}_i}) - \left[\sum_{\boldsymbol{x}_j \in \mathcal{C}_j} d(\boldsymbol{x}_j, \boldsymbol{\mu}_{\mathcal{C}_j}) + \sum_{\boldsymbol{x}_j \in \mathcal{C}_i} d(\boldsymbol{x}_j, \boldsymbol{\mu}_{\mathcal{C}_i}) \right], \tag{2.6.4}$$

where $\boldsymbol{\mu}_{\mathcal{C}_j}$ denotes the centroid of the cluster $\mathcal{C}_j$.

This iterative agglomerative clustering process can furthermore be represented by a tree structure, called **dendrogram**. It shows how the data observations are clustered together step-by-step and height values of the tree correspond to a similarity between clusters such as in Equation (2.6.3) or (2.6.4). As an example, we create six observations in $\mathbb{R}^2$ as shown in Figure 2.6.1 and apply the agglomerative clustering with average linkage and Euclidean distance, which are defined in Equations (2.6.3) and (2.6.2), respectively. Figure

2.6.2 presents the resulting dendrogram, where all observations are singleton clusters at height 0.0. At the height 0.5, the observations $\boldsymbol{x}_1$ and $\boldsymbol{x}_3$ are grouped together to form the first cluster and they stay together at all next levels. Note that the value 0.5 is the average linkage value between $\boldsymbol{x}_1$ and $\boldsymbol{x}_3$ with Euclidean distance. Moreover, when the similarity of two groups of observation, $\{\boldsymbol{x}_1, \boldsymbol{x}_2, \boldsymbol{x}_3\}$ and $\{\boldsymbol{x}_4, \boldsymbol{x}_5, \boldsymbol{x}_6\}$, is just above 1.0 in this data set, they are merged together to form a single cluster.

On the other hand, as can be seen, the agglomerative clustering can neither undo what was done previously nor perform the observation swapping between clusters, which is the major difficulty by obtaining high-quality clusters.

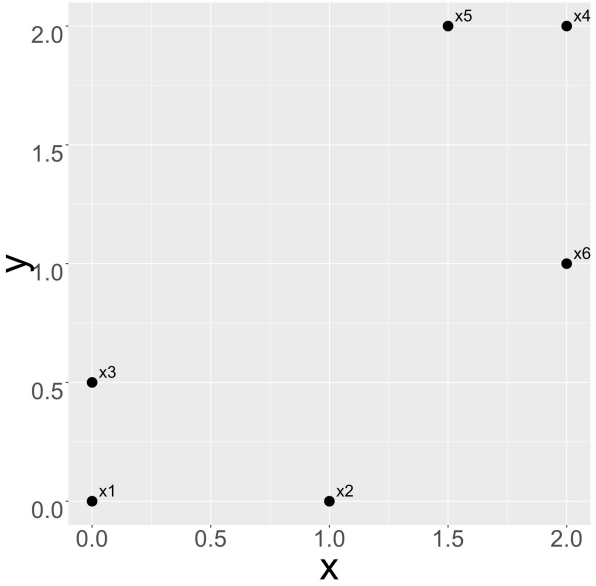


Figure 2.6.1: Data observations to cluster.

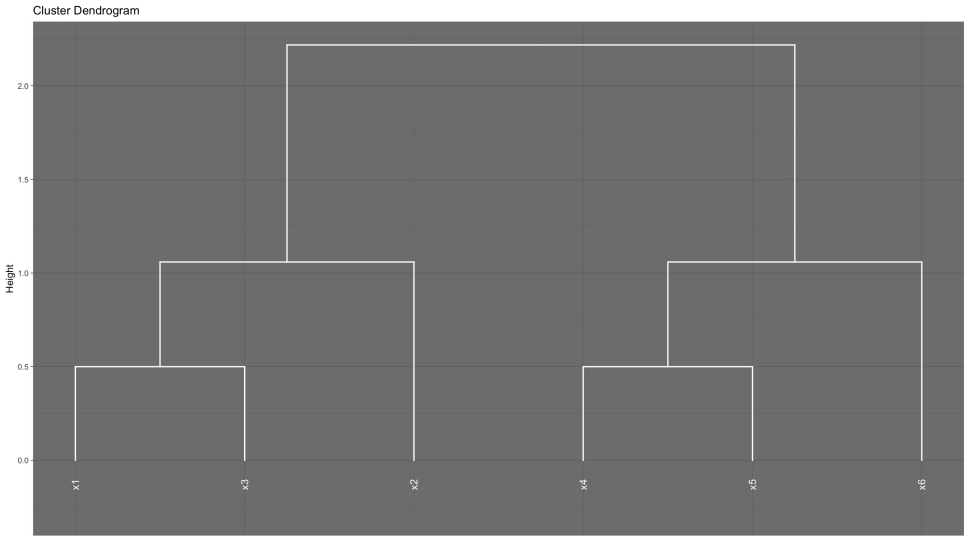


Figure 2.6.2: The dendrogram obtained applying the agglomerative clustering on the data shown in Figure 2.6.1.

Algorithm 2 Agglomerative hierarchical clustering algorithm

Input: A set of data observations $X = \{\boldsymbol{x}_1, ..., \boldsymbol{x}_n\} \subset \mathbb{R}^d$, a distance function $d : (X, X) \to \mathbb{R}$ such as in Equation (2.6.3) or (2.6.4) and the initial clusters with $\mathcal{C}_j := \{\boldsymbol{x_j}\}$ for $j = 1, ..., n$.
Output: The clustering hierarchy.
$\mathcal{C} = \{\mathcal{C}_1, ..., \mathcal{C}_k\}$
while $|\mathcal{C}| \neq 1$ **do**
    $(\mathcal{C}_+, \mathcal{C}_{++}) := \operatorname{argmin}_{\substack{c_i, c_j \in \mathcal{C} \\ \mathcal{C}_i \neq \mathcal{C}_j}} d(\mathcal{C}_i, \mathcal{C}_j)$
    $\mathcal{C}^{new} := \mathcal{C}_+ \cup \mathcal{C}_{++}$
    $\mathcal{C}^{old} := \mathcal{C} \setminus \{\mathcal{C}_+, \mathcal{C}_{++}\}$
    $\mathcal{C} = \mathcal{C}^{new} \cup \mathcal{C}^{old}$
end while

2.6.3 Validation indices

One of the major difficulties in clustering is to specify the number clusters k. As we have seen in Section 2.6.2, partitioning clustering methods usually need to specify the number of clusters k beforehand, while for hierarchical methods the value of k can be determined afterwards. The number of clusters k is usually an unknown parameter, which has to be set based on prior knowledge (**external cluster criterion**) or based on an estimation (**internal cluster criterion**). Because we do not have any external information about the class labels or any prior knowledge about the data in the thesis, we use the internal cluster criteria.

Although there have been many different internal cluster criteria to estimate the optimal number of clusters in the literature, we follow the results of the study carried out by Milligan et al. (1985), where they compared 30 criteria that estimate the number of clusters. They performed Monte Carlo simulations and created artificial test data sets for which the optimal numbers of clusters are known. Next, they assessed performance of each method. As a result, they reported the five indices that correctly estimated the number of clusters most frequently and we select one of them to apply in the thesis, gamma index. On the other hand, silhouette index is another criteria that one encounters in many implementations and articles; thus, it is our second index while setting the number of clusters k.

In the following parts we explain these two indices, the silhouette index and the gamma index.

Silhouette index

The silhouette index evaluates a clustering based on how well the clusters are separated and how compact the clusters are, which is proposed by Rousseeuw (1987). We give its formulation in Equation (2.6.7).

Definition 2.6.6. *(Silhouette index)*
Let $X = \{\boldsymbol{x}_1, ..., \boldsymbol{x}_n\} \subset \mathbb{R}^d$ denote a set of observations and $\mathcal{C} = \{\mathcal{C}_1, ..., \mathcal{C}_k\}$ denote its partition into k clusters. Moreover, define a distance function $d : (X, X) \to \mathbb{R}$. For each object $\boldsymbol{x} \in X$ we calculate $a(\boldsymbol{x})$ as the average distance between $\boldsymbol{x}$ and all other observations in the cluster to which $\boldsymbol{x}$ belongs. Similarly, $b(\boldsymbol{x})$ is the minimum average distance from $\boldsymbol{x}$ to all clusters to which $\boldsymbol{x}$ does not belong. Suppose $\boldsymbol{x} \in \mathcal{C}_j$ $(1 \leq j \leq k)$; then $a(\boldsymbol{x})$ and $b(\boldsymbol{x})$ can be written as:

$$a(\boldsymbol{x}) := \frac{\sum_{\substack{\boldsymbol{x}_j \in \mathcal{C}_j \\ \boldsymbol{x}_j \neq \boldsymbol{x}}} d(\boldsymbol{x}_j, \boldsymbol{x})}{|\mathcal{C}_j| - 1} \ , \tag{2.6.5}$$

$$b(\boldsymbol{x}) := \min_{\substack{\mathcal{C}_i \in \mathcal{C} \\ 1 \leq i \leq k \\ i \neq j}} \left\{ \frac{\sum_{\boldsymbol{x}_i \in \mathcal{C}_i} d(\boldsymbol{x}_i, \boldsymbol{x})}{|\mathcal{C}_i|} \right\} \ . \tag{2.6.6}$$

*Then the **silhouette index** of $\boldsymbol{x}$ is defined as*

$$s(\boldsymbol{x}) := \frac{b(\boldsymbol{x}) - a(\boldsymbol{x})}{\max\{a(\boldsymbol{x}), b(\boldsymbol{x})\}} \ . \tag{2.6.7}$$

From the definition it holds that $a(\boldsymbol{x})$ and $b(\boldsymbol{x})$ are positive. If we have the case of $0 \leq b(\boldsymbol{x}) \leq a(\boldsymbol{x})$, the denominator $(\max\{a(\boldsymbol{x}), b(\boldsymbol{x})\})$ returns $a(\boldsymbol{x})$. Since $b(\boldsymbol{x}) - a(\boldsymbol{x}) \geq -a(\boldsymbol{x})$ holds, the minimum value of the silhouette index is -1. As the second case we can have $0 \leq a(\boldsymbol{x}) \leq b(\boldsymbol{x})$. Then the denominator returns $b(\boldsymbol{x})$. Because $b(\boldsymbol{x}) - a(\boldsymbol{x}) \leq b(\boldsymbol{x})$ holds, the maximum value of the silhouette index is 1. As a result, it holds $s(\boldsymbol{x}) \in [-1, 1]$. Moreover, the value that we defined in Equation (2.6.5) reflects the compactness of the cluster to which the observation belongs. The smaller the value, the more compact the cluster. On the other hand, what we formulated in Equation (2.6.6), is the separation value of the cluster. The larger the value is, the more separated the observation is from other clusters. Hence, a silhouette value close to 1 shows that the observation belongs to the appropriate cluster, whereas a negative silhouette value implies that the observation should be in another cluster.

To measure the quality of a clustering, the average silhouette value of all observations in the data set can be calculated.

Gamma index

The gamma index, known as Baker-Hubert gamma index, evaluates a clustering based on the distances between and within the clusters as proposed by Baker et al. (1975). Its formulation can be seen in Equation (2.7.7).

Definition 2.6.7. *(Gamma index)*
Let $X = \{\boldsymbol{x}_1, ..., \boldsymbol{x}_n\} \subset \mathbb{R}^d$ denote a set of observations and $\mathcal{C} = \{\mathcal{C}_1, ..., \mathcal{C}_k\}$ denote its partition into k clusters. Moreover, define a distance function $d : (X, X) \to \mathbb{R}$. For each pair of observations we calculate S^+ as the number of times a pair of observations that

belong to the same cluster has a smaller distance than two observations that belong to different clusters. Conversely, S^- is how many times a pair of observations that belong to the same cluster has a larger distance than two observations that belong to different clusters. For $\boldsymbol{x}_a, \boldsymbol{x}_b, \boldsymbol{x}_c, \boldsymbol{x}_e \in X$ the formal definitions of S^+ and S^- can be written as:

$$S^+ := \frac{1}{2}\sum_{j=1}^{k} \sum_{\substack{\boldsymbol{x}_a, \boldsymbol{x}_b \in \mathcal{C}_j \\ a \neq b}} \frac{1}{2}\sum_{i=1}^{k} \sum_{\substack{\boldsymbol{x}_c \in \mathcal{C}_i \\ \boldsymbol{x}_e \notin \mathcal{C}_i}} \mathbf{1}_{\{d(\boldsymbol{x}_a, \boldsymbol{x}_b) < d(\boldsymbol{x}_c, \boldsymbol{x}_e)\}} \ , \tag{2.6.8}$$

$$S^- := \frac{1}{2}\sum_{j=1}^{k} \sum_{\substack{\boldsymbol{x}_a, \boldsymbol{x}_b \in \mathcal{C}_j \\ a \neq b}} \frac{1}{2}\sum_{i=1}^{k} \sum_{\substack{\boldsymbol{x}_c \in \mathcal{C}_i \\ \boldsymbol{x}_e \notin \mathcal{C}_i}} \mathbf{1}_{\{d(\boldsymbol{x}_a, \boldsymbol{x}_b) > d(\boldsymbol{x}_c, \boldsymbol{x}_e)\}} \ . \tag{2.6.9}$$

*Then the **gamma index** can be calculated by*

$$\Gamma := \frac{S^+ - S^-}{S^+ + S^-} = 1 - \frac{2S^-}{S^+ + S^-} \ . \tag{2.6.10}$$

Note that Equation (2.6.8) and (2.6.9) are strict, thus the cases where the distances between pairs $(\boldsymbol{x}_a, \boldsymbol{x}_b)$ and $(\boldsymbol{x}_c, \boldsymbol{x}_e)$ are equal are not considered. The factors $\frac{1}{2}$ are added to avoid counting pairs twice. From the definition it holds that both S^+ and S^- are nonnegative. If we consider the case of $0 \leq S^- \leq S^+$, then the dominator is bigger than or equal to the nominator, i.e. $2S^- \leq S^+ + S^-$. Thus the maximum value of the gamma index is 1. As the second case we can consider $0 \leq S^+ \leq S^-$. Then it holds $\frac{2S^-}{S^+ + S^-} < 2$. Hence, the minimum value of the gamma index is -1. As a result, what Equation (2.7.7) implies, is that the gamma index is between -1 and 1.

The higher values of S^+ and lower values of S^+ imply higher Γ values thereby a good clustering. Thus, one should calculate Γ for a range of values of the number of clusters k and select k that maximizes the index.

2.7 Copulas

In the previous parts we defined distributions, data preprocessing steps, models, performance measures and clustering methods we are using. In this section we introduce copulas, a popular model class for analyzing multivariate data. Let $\boldsymbol{X} = (X_1, ..., X_d)^\top$ be a d-dimensional vector of continuous random variables and F denote its joint distribution function. Moreover, let f denote its joint density function. Suppose F_j defines the corresponding marginal distribution functions and f_j denotes the marginal density functions for $j = 1, ..., d$. Then, the joint distribution function includes information both on the individual behavior of the random variables and on the dependence between them. Nonetheless, estimating the joint distribution function, F, can be difficult in higher dimensions and copulas help us to deal with this problem in a flexible way. In the following, we give more details about copulas and the presentation in this section are mostly from Czado (2019) if not stated otherwise.

2.7.1 Definition and basic properties

Definition 2.7.1. *(PIT)*
Let $X \in \mathbb{R}$ be a continuous random variable with distribution function F and let x be an observed value of X. Then the ***probability integral transform (PIT)*** *is defined by $u := F(x)$.*

Moreover, if $X \sim F$ holds, then $\mathrm{U} = F(X)$ has a uniform distribution, that we can show as follows:

$P(U \leqslant u) = P(F(X) \leqslant u) = P(X \leqslant F^{-1}(u)) = F(F^{-1}(u)) = u$ holds for every $u \in [0, 1]$.

As a result, we can apply the PIT to marginal distributions of our interest $X = (X_1, ..., X_d)^\top$ and obtain uniformly distributed data $U = (U_1, ..., U_d)^\top$, whose dependencies can be formulated later by joint distributions. Meaning that, we have

$$U_j = F_j(X_j) \quad \text{for} \quad j = 1, ..., d. \tag{2.7.1}$$

Moreover, the resulting data is called as in **u-scale** or **copula-scale**.

Definition 2.7.2. *(Copula)*
A d-dimensional ***copula*** *C is a multivariate distribution function*

$$C : [0, 1]^d \to [0, 1],$$

where all marginals are uniformly distributed on $[0, 1]$.

Definition 2.7.3. *(Copula density)*
Let C be a d-dimensional copula. Then the corresponding ***copula density*** *c is given by*

$$c(u_1, ..., u_d) := \frac{\partial^d}{\partial u_1 ... \partial u_d} C(u_1, ..., u_d) \quad \textit{for} \quad u \in [0, 1]^d \ . \tag{2.7.2}$$

Now we present one of the major theorems of copulas proven by Sklar (1959). It allows us to specify multivariate distributions in terms of their marginals and a corresponding copula.

Theorem 2.7.1. *(Sklar's Theorem)*
Let $\boldsymbol{X} = (X_1, ..., X_d)^\top \in \mathbb{R}^d$ *be a d-dimensional random vector with continuous joint distribution function* $F : \mathbb{R}^d \to [0, 1]$ *and marginal distribution functions* $F_1, ..., F_d : \mathbb{R} \to [0, 1]$. *Then the joint distribution function can be written as*

$$F(x_1, ..., x_d) = C(F_1(x_1), ..., F_d(x_d)). \tag{2.7.3}$$

Moreover, for the associated density we get

$$f(x_1, ..., x_d) = c(F_1(x_1), ..., F_d(x_d))f_1(x_1)...f_d(x_d), \tag{2.7.4}$$

where c denotes the probability density function of the d-dimensional copula C and $f_1, ..., f_d$ *are the marginal density functions of* $X_1, ..., X_d$.

The inverse statement holds as well; thus, the copula corresponding to a multivariate distribution function $F : \mathbb{R}^d \to [0, 1]$ *with marginal distribution functions* $F_1, ..., F_d : \mathbb{R} \to [0, 1]$ *can be expressed as*

$$C(u_1, ..., u_d) = F(F_1^{-1}(u_1), ..., F_d^{-1}(u_d)) \tag{2.7.5}$$

and its copula density is given by

$$c(u_1, ..., u_d) = \frac{f(F_1^{-1}(u_1), ..., F_d^{-1}(u_d))}{f_1(F_1^{-1}(u_1))...f_d(F_d^{-1}(u_d))}. \tag{2.7.6}$$

Theorem 2.7.2. *(Invariance of copulas)*
Let H_j*'s be invertible and strictly increasing functions for* $j = 1, ..., d$ *and* $\boldsymbol{X} = (X_1, ..., X_d)^\top$ *denote a random vector. Furthermore, define* $Y_j = H_j(X_j)$ *for* $j = 1, ..., d$. *Then the copula of* $\boldsymbol{Y} = (Y_1, ..., Y_d)^\top$ *represented by* C_Y *agrees with the copula* C_X *of* X.

Proof The proofs can be found in the book by Nelsen (2006).

2.7.2 Bivariate copula families

After defining copulas and stating its essential properties, we now present a number of commonly used copula families. For simplicity we restrict our examples to the bivariate case, which includes both non-parametric and parametric copulas. We firstly introduce several nonparametric copulas as follows:

Example 2.7.1. *(Bivariate independence copula)*
Let U_1 *and* U_2 *be independent uniformly distributed random variables on the interval* $[0, 1]$. *Then the* ***bivariate independence copula*** *is given by*

$$P(U_1 \leq u_1, U_2 \leq u_2) = u_1 u_2. \tag{2.7.7}$$

Example 2.7.2. *(Bivariate comonotonicity copula)*
Let U_1 and U_2 be independent uniformly distributed random variables on the interval $[0, 1]$. Then the ***bivariate comonotonicity copula*** *is given by*

$$P(U_1 \leq u_1, U_2 \leq u_2) = \min\{u_1, u_2\}. \tag{2.7.8}$$

It corresponds to the case of extreme positive dependence.

Example 2.7.3. *(Countermonotonicity copula)*
Let U_1 and U_2 be independent uniformly distributed random variables on the interval $[0, 1]$. Then the ***countermonotonicity copula*** *is given by*

$$P(U_1 \leq u_1, U_2 \leq u_2) = \max\{u_1 + u_2 - 1, 0\}. \tag{2.7.9}$$

It corresponds to the case of perfect negative dependence, thus it is the bivariate copula by definition.

Subsequently, we discuss the parametric copula families, which can be constructed by different approaches. For instance, using any arbitrary parametric distribution function F in the inverse statement (2.7.5) of Sklar's Theorem can result in a parametric copula family. Moreover, multivariate extreme-value theory and generator functions help us to define new copulas.

We first of all discuss the parametric copulas derived from applying the inverse statement (2.7.5) of Sklar's Theorem to the elliptical distributions such as bivariate normal and Student's t distributions. What we obtain, is two important elliptical copula families, Gaussian and Student's t copula.

Example 2.7.4. *(Bivariate Gaussian copula)*
To construct the ***bivariate Gaussian copula****, one can use a bivariate normal distribution with zero mean vector, unit variances and correlation ρ and apply the inverse statement (2.7.5) of Sklar's Theorem, i.e.*

$$C(u_1, u_2; \rho) := \Phi_2(\Phi_1^{-1}(u_1), \Phi_1^{-1}(u_2); \rho)\ , \tag{2.7.10}$$

where $\Phi_1^{-1}(\cdot)$ is the inverse of the univariate standard normal distribution function and $\Phi_2(\cdot\,, \cdot; \rho)$ is the bivariate normal distribution function with zero means, unit variances and correlation ρ.

Example 2.7.5. *(Bivariate Student's t copula)*
Applying the inverse statement (2.7.5) of Sklar's Theorem to a bivariate Student's t distribution with zero mean vector, correlation ρ and degrees of freedom $\nu > 0$ results in the ***bivariate Student's t copula****, which can be formulated as:*

$$C(u_1, u_2; \nu, \rho) := T_{2,\nu}(T_\nu^{-1}(u_1), T_\nu^{-1}(u_2); \nu, \rho)\ , \tag{2.7.11}$$

where $T_\nu^{-1}(\cdot)$ is the inverse of the univariate Student's t distribution function with degrees of freedom $\nu > 0$ and $T_{2,\nu}(\cdot\,, \cdot; \nu, \rho)$ is the bivariate Student's t distribution function with zero means, degrees of freedom $\nu > 0$ and correlation ρ.

Another well-known copula families apart from the elliptical copulas belong to Archimedean copulas, where one uses generator functions to obtain copulas. Now we briefly introduce the bivariate Archimedean copulas and give several examples of this class.

Definition 2.7.4. *(Bivariate Archimedean copula)*
Let Ω be the set of all continuous, strictly monotone decreasing and convex functions $\varphi : I \to [0, \infty]$ with $\varphi(1) = 0$. Assume $\varphi \in \Omega$, then

$$C(u_1, u_2) := \varphi^{[-1]}(\varphi(u_1) + \varphi(u_2)) \ , \tag{2.7.12}$$

*is a **bivariate Archimedean copula** with a **generator** φ. Here $\varphi^{[-1]}$ is the pseudo-inverse of φ, defined as $\varphi^{[-1]} : [0, \infty] \to [0, 1]$ with*

$$\varphi^{[-1]}(x) := \begin{cases} \varphi^{-1}(x) & for \ \ 0 \leq x \leq \varphi(0) \\ 0 & for \ \ \varphi(0) \leq x \leq \infty. \end{cases} \tag{2.7.13}$$

What Equation (2.7.12) and (2.7.13) imply, is that we can construct bivariate Archimedean copulas with appropriate choices for a generator function φ. In the following examples some important copulas of this class with their generator function can be seen.

Example 2.7.6. *(Bivariate Clayton copula)*
Let the generator function φ be given by

$$\varphi(x) = \tfrac{1}{\delta}(x^{-\delta} - 1),$$

*where $\delta \in (0, \infty)$ denotes the dependence parameter. Then the **bivariate Clayton copula** is defined as*

$$C(u_1, u_2; \delta) := (u_1^{-\delta} + u_2^{-\delta} - 1)^{-\frac{1}{\delta}} \ . \tag{2.7.14}$$

Example 2.7.7. *(Bivariate Gumbel copula)*
Let the generator function φ be given by

$$\varphi(x) = (-\log(x))^{\delta},$$

*where $\delta \in [1, \infty]$ denotes the dependence parameter. Then the **bivariate Gumbel copula** is defined as*

$$C(u_1, u_2; \delta) := exp\left\{ -\left[(-\log(u_1))^{\delta} + (-\log(u_2))^{\delta} \right]^{\frac{1}{\delta}} \right\} \ . \tag{2.7.15}$$

Example 2.7.8. *(Bivariate Frank copula)*
Let the generator function φ be given by

$$\varphi(x) = -\log\left(\tfrac{exp(-\delta x)-1}{exp(-\delta)-1} \right),$$

where $\delta \in \mathbb{R} \setminus \{0\}$ denotes the dependence parameter. Then the ***bivariate Frank copula*** *is defined as*

$$C(u_1, u_2; \delta) := -\frac{1}{\delta} \log \left(1 + \frac{(exp(-\delta u_1) - 1)\,(exp(-\delta u_2) - 1)}{exp(-\delta) - 1} \right) \ . \tag{2.7.16}$$

Example 2.7.9. *(Bivariate Joe copula)*
Let the generator function φ be given by

$$\varphi(x) = -\log\big(1 - (1 - x)^{\delta}\big),$$

where $\delta \in [1, \infty]$ denotes the dependence parameter. Then the ***bivariate Joe copula*** *is defined as*

$$C(u_1, u_2; \delta) := 1 - \big((1 - u_1)^{\delta} + (1 - u_2)^{\delta} - (1 - u_1)^{\delta}(1 - u_2)^{\delta}\big)^{\frac{1}{\delta}} \ . \tag{2.7.17}$$

All the Archimedean copulas that we described above have one parameter. There are additionally the Archimedean copulas available with two parameters such as the BB1 and BB7 copulas. We are not presenting them, but more information on other copulas can be found in the references given at the beginning of the section.

2.7.3 Dependence measures

There are a number of measures for the strength and direction of the dependence between two random variables. In this section we present two of them: **Kendall's tau**, developed by Kendall (1938) and **tail dependence**, a property of the copula.

Definition 2.7.5. *(Kendall's τ)*
Let (X_1, X_2) be two continuous random variables and (X_{11}, X_{12}), (X_{21}, X_{22}) be independent and identically distributed copies of (X_1, X_2). Then the ***Kendall's τ*** *between (X_1, X_2) is given by*

$$\tau(X_1, X_2) := \mathbb{P}((X_{11} - X_{21})(X_{12} - X_{22}) > 0) - \mathbb{P}((X_{11} - X_{21})(X_{12} - X_{22}) < 0) \ . \tag{2.7.18}$$

Next, we discuss the estimation of Kendall's τ and for this goal we firstly define concordance and discordance.

Definition 2.7.6. *(Concordance and discordance)*
Let (X_1, X_2) be two continuous random variables and (x_{j1}, x_{j2}), (x_{k1}, x_{k2}) be realizations of (X_1, X_2) for $j, k = 1, ..., n$, where n is the sample size. Then the pair (x_{j1}, x_{j2}) and (x_{k1}, x_{k2}) is called ***concordant****, if it holds*

$$(x_{j1} - x_{k1})(x_{i2} - x_{j2}) > 0 \tag{2.7.19}$$

and ***discordant*** *if we have*

$$(x_{j1} - x_{k1})(x_{j2} - x_{k2}) < 0. \tag{2.7.20}$$

Moreover, we call it as ***extra*** x_1 ***pair*** *or* ***extra*** x_2 ***pair****, if*

$$x_{j1} = x_{k1} \quad or \quad x_{j2} = x_{k2}, \tag{2.7.21}$$

respectively.

Definition 2.7.7. *(Kendall's τ estimate)*
Let (X_1, X_2) be two continuous random variables and (x_{j1}, x_{j2}) and (x_{k1}, x_{k2}) be realizations of (X_1, X_2) for $j, k = 1, ..., n$, where n is the sample size. Moreover, let N_c denote the number of concordant pairs, N_d denote the number of discordant pairs, N_1 denote the number of extra x_1 pairs and N_2 denote the number of extra x_2 pairs. Then an ***empirical estimate of the Kendall's*** *τ between them is defined as*

$$\hat{\tau}(X_1, X_2) := \frac{N_c - N_d}{\sqrt{N_c + N_d + N_1}\sqrt{N_c + N_d + N_2}}. \tag{2.7.22}$$

In the following Lemma, we summarize the most important properties of Kendall's τ.

Lemma 2.7.3. *(Properties of Kendall's τ)*
Let (X_1, X_2) be two random variables with joint distribution F and copula C. Then,

1. $\tau(X_1, X_2) \in [-1, 1]$.
2. *If X_1 and X_2 are independent, $\tau(X_1, X_2) = 0$.*
3. $\tau(X_1, X_2) = 4 \int_{[0,1]^2} C(u_1, u_2) dC(u_1, u_2) - 1$. *As can be seen, Kendall's τ is invariant with respect to monotone transformations of the margins and thus, is independent of marginal effects.*

For many copula families, there is a bijection between the copula parameter and Kendall's τ. In Table 2.7.1 a summary of the bijections can be seen for some of the discussed bivariate copula families.

Family	Kendall's τ	Range of τ
Gaussian	$\frac{2}{\pi} \arcsin(\rho)$	$[-1, 1]$
Student's t	$\frac{2}{\pi} \arcsin(\rho)$	$[-1, 1]$
Clayton	$\frac{\delta}{\delta+2}$	$[0, 1]$
Gumbel	$1 - \frac{1}{\delta}$	$[0, 1]$
Frank	$1 + 4(\frac{D(\delta)-1}{\delta})$ with $D(\delta) = \int_0^\delta \frac{x/\delta}{exp(x)-1} dx$	$[-1, 1]$
Joe	$1 + \frac{\left(-2+2\gamma+2\log(2)+\Psi(\frac{1}{\delta})+\Psi(\frac{2+\delta}{2\delta})+\delta\right)}{(\delta-2)}$ with Euler's constant $\gamma \approx 0.57721$ and Digamma-function Ψ	$[0, 1]$

Table 2.7.1: Kendall's τ in terms of the copula parameters of selected bivariate copula families.

Note that some of the discussed copula families only allow for positive dependence, $(\tau > 0)$. How this range can be extended to have more flexibility, is rotation of the copula.

Definition 2.7.8. *(Rotated copulas)*
Let $c(u_1, u_2)$ be a copula density. Then the densities of counter-clockwise rotated versions of this copula are given by

1. *90 degrees rotation: $c_{90}(u_1, u_2) := c(1 - u_1, u_2)$.*
2. *180 degrees rotation: $c_{180}(u_1, u_2) := c(1 - u_1, 1 - u_2)$.*
3. *270 degrees rotation: $c_{270}(u_1, u_2) := c(u_1, 1 - u_2)$.*

As a second measure, we discuss the tail dependence, where the dependence between extreme values of two random variables are considered.

Definition 2.7.9. *(Tail-dependence coefficients)*
Let (X_1, X_2) be two random variables and $X_1 \sim F_1$, $X_2 \sim F_2$ hold. Assume we investigate the dependence between the extreme high values of X_1 and X_2. Then the ***upper tail-dependence coefficient*** *is defined as*

$$\lambda_U(X_1, X_2) := \lim_{u \to 1^-} P(X_1 > F_1^{-1}(u) | X_2 > F_2^{-1}(u)), \tag{2.7.23}$$

provided it exists. Similarly, we can define the ***lower tail-dependence coefficient*** *for the dependence between the extreme low values of X_1 and X_2 as*

$$\lambda_L(X_1, X_2) := \lim_{u \to 0^+} P(X_1 \leq F_1^{-1}(u) | X_2 \leq F_2^{-1}(u)), \tag{2.7.24}$$

provided it exists.

It can be shown that the tail-dependence is a copula property and its definition in terms of copula is as follows.

Definition 2.7.10. *(Copula tail-dependence coefficients)*
Let C be a bivariate copula of a random vector (X_1, X_2). Then the ***upper and lower tail-dependence coefficients*** *are defined as*

$$\lambda_U(X_1, X_2) := \lim_{u \to 1^-} \frac{1 - 2u + C(u, u)}{1 - u}, \tag{2.7.25}$$

$$\lambda_L(X_1, X_2) := \lim_{u \to 0^+} \frac{C(u, u)}{u}, \tag{2.7.26}$$

provided they exist.

Whenever λ_U / λ_L is bigger than zero, we can state that X_1 and X_2 are upper/lower tail-dependent. Moreover, because the tail-dependence is a copula property, there is an explicit relationship between the copula parameters and the tail-dependence coefficients. In Table 2.7.2 a summary of the relationship can be seen for some of the discussed bivariate copula families.

Family	λ_U	λ_L
Gaussian	0	0
Student's t	$2t_{\nu+1}\left(-\sqrt{\nu+1}\sqrt{\frac{1-\rho}{1+\rho}}\right)$	$2t_{\nu+1}\left(-\sqrt{\nu+1}\sqrt{\frac{1-\rho}{1+\rho}}\right)$
Clayton	0	$2^{-1/\delta}$
Gumbel	2 - $2^{1/\delta}$	0
Frank	0	0
Joe	2 - $2^{1/\delta}$	0

Table 2.7.2: Tail-dependence coefficients in terms of the copula parameters of selected bivariate copula families.

2.7.4 Exploratory visualization

In the previous sections we defined the copula and some of important dependence measures. What we additionally discussed, is that many bivariate copula families have relationships with these measures. Because having a good visualization helps us to understand the bivariate copula families and their relationships, in this section we explain the most common visualization tools for the bivariate copulas.

We start with the **scatter plot**, the most popular visualization method of copula samples. Since the support of the copula is the unit square, in the scatter plot each observation is represented by a point in the unit square. As an example, what we see in Figure 2.7.1, is the scatter plots of the simulated data observations of selected parametric bivariate copula families. The sample size is 500 and we choose the parameters so that the weak ($\tau = 0.2$) and strong ($\tau = 0.8$) dependence can be observed. Moreover, it is easy to see that different tail dependencies and asymmetries exist for the Clayton and Gumbel copulas. On the other hand, it may be hard to interpret these plots in different scenarios and we cannot select the appropriate copula family for the data.

The tool that overcomes the problem of the scatter plot is the **normalized copula contour plot**, where the copula is coupled with the standard normal margins. Thus, instead of inspecting the original copula density, we look at the transformed density given by

$$g(z_1, z_2) := c(\Phi(z_1), \Phi(z_2))\phi(z_1)\phi(z_2), \tag{2.7.27}$$

where $\Phi(\cdot)$ is the univariate standard normal distribution function and $\phi(\cdot)$ is the univariate standard normal density function. How we use it in the visual inspection, is that we draw contour plots of this density and analyze the width of the contours for the strength of dependence or the shape in tails for the tail-dependence. The normalized copula contour plots of the previous copulas provided in Figure 2.7.1 can be seen in Figure 2.7.2. We observe that the upper tail dependence can be modeled by either the Gumbel or the Joe copula and the lower tail dependence similarly can be represented by the Clayton copula, whereas the Frank and the Gaussian copula families do not have any tail-dependence property. Moreover, the Student's t copula helps us to model both the lower and upper tail dependence at the same time. What we additionally see in Figure 2.7.2, is that wider

the contours, the weaker the dependence.

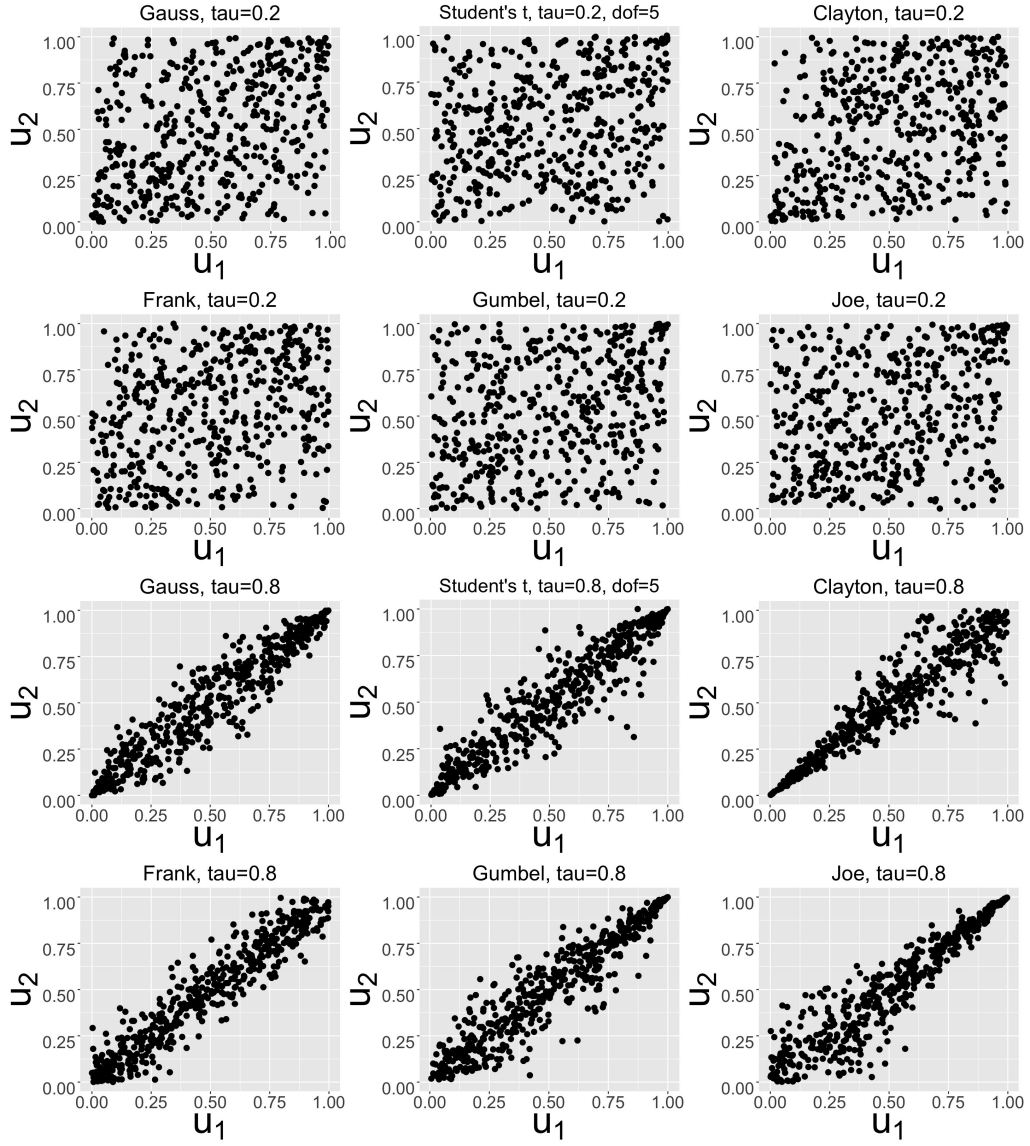


Figure 2.7.1: Scatter plots of simulated copula data for weak (τ=0.2) and strong (τ=0.8) dependence. Sample size is 500.

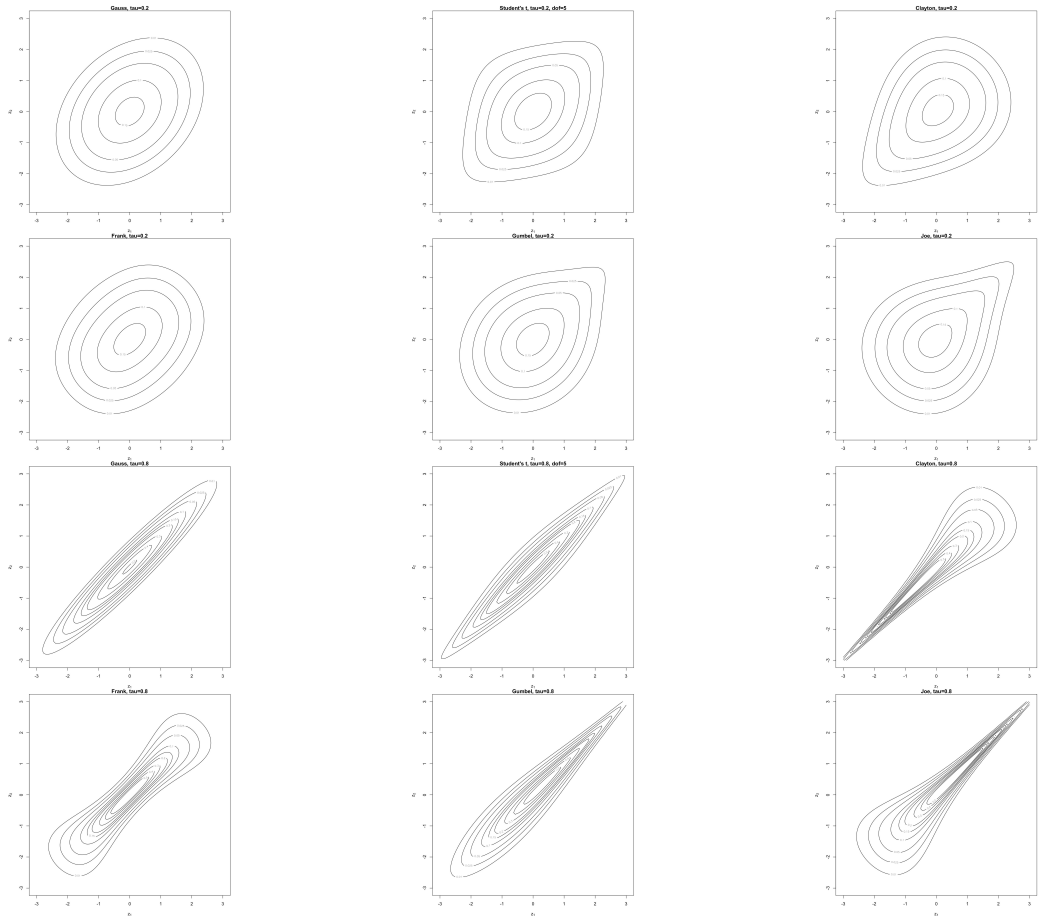


Figure 2.7.2: Marginal normal contour plots of copula densities for weak (τ=0.2) and strong (τ=0.8) dependence.

2.7.5 Bivariate parametric copula estimation

Let (u_{j1}, u_{j2}), $j = 1, ..., n$, be i.i.d. samples of a random vector $(U_1, U_2) \sim C$ and assume we want to the estimate the bivariate copula C. The most popular method in parametric copula models is to assume a particular family and estimate its parameter(s) by maximum likelihood.

Definition 2.7.11. *(Maximum likelihood estimator of the bivariate copula parameter) Assume we have $(U_1, U_2) \sim C(.\,, .; \theta)$, where $\theta \in \Theta$ and $\Theta \subset \mathbb{R}^p$ ($p \in \mathbb{N}$) is the parameter space of the family. Let $c(.\,, .; \theta)$ denote the density of $C(.\,, .; \theta)$. Then the* ***maximum likelihood estimator*** *of the parameter θ is given by*

$$\hat{\theta}_n^{MLE} = \text{argmax}_{\theta \in \Theta} \prod_{j=1}^{n} c(u_{j1}\,, u_{j2}; \theta)\,. \tag{2.7.28}$$

To estimate the parameters of the copula family, we should choose the parametric family. Usually, the parameter is estimated for several different parametric families and the best

model is selected by considering information criteria. The most popular criteria is AIC that we defined in Equation (2.5.4). The best model is the one that minimizes the AIC. When the number of parameters across the considered models is the same, the criteria selects the model that gives the highest likelihood.

Chapter 3

Data set

After explaining some methods and models in Section 2, we now apply them to a real-world data set. Besides, we compare the performance of all models in Section 4.1. For our application we consider the data set from a German pharmacy software company. They operate all over Germany and currently there are 2, 695 pharmacies in their system.

3.1 Introduction to data set

Our data contains weekly number of sales of the most sold 10 products from January 10th, 2016 until March 31st, 2019. As a result, the total number of weeks in the data is 169 and each record represents the weekly number of sales of one out of 2, 695 pharmacies. The software company provides, further, the first two digits of ZIP codes of the pharmacies. Therefore, we group the pharmacies according to these digits and obtain 95 different groups (regions), which we abbreviate by `pl`. Then, we sum up the weekly number of sales of the whole pharmacies, which belong to the same region, and have total weekly number of sales per `pl` region. Finally, we analyze the total weekly number of sales of 95 `pl` regions.

We give a variable description including response and explanatory variables in Table 3.1.1. Here we abbreviate the indices of products $p = 1, ..., 10$, of regions $j = 1, ..., 95$ and of weeks $t = 1, ..., 169$. For more details about third party data, see Section 3.2.

Here we point out that the covariates changing only with regions such as doc_j and $power_j$ do not affect the model fitting in case we fix the region in the model setup. In other words, the response, y_{tjp} with j fixed, depends on the week t and product p; therefore, on the covariates like $temp_{tj}$ or $hreg_{tp}$.

Variable	Measurement unit	Variable type	Domain	Description
Response				
$log(y_{tjp})$	Package	Continuous	$\mathbb{R}_+$	Logarithm of total sales of product p at week t in region j
Covariates				
Time varying				
$temp_{tj}$	Celsius	Continuous	$\mathbb{R}$	Mean temperature at week t in region j
hol_{tj}	Day	Discrete	$\mathbb{R}_+$	Total school holidays at week t in region j
$google_{tjp}$	View	Continuous	$\mathbb{R}_{\geq 0}$	Google keyword search at week t for product p in region j
$hreg_{tp}$	-	Continuous	$\mathbb{R}$	Harmonic regression estimate at week t for product p
Time independent				
doc_j	Person	Continuous	$\mathbb{R}_+$	Total number of doctors in region j
$hosp_j$	-	Continuous	$\mathbb{R}_+$	Total number of hospitals in region j
$power_j$	-	Discrete	$\mathbb{R}_+$	Purchasing power in region j
$store_j$	-	Discrete	$\mathbb{R}_+$	Total number of pharmacies in region j
pop_j	Person	Continuous	$\mathbb{R}_+$	Total number of inhabitants in region j

Table 3.1.1: Description of variables in the sales and third party data.

3.2 Third party data

As can be seen in Section 3.1.1, we supplement our data with third party data from various sources:

1. The mapping of latitude and longitude into ZIP codes and regions, extracted on June 2019, according to
   https://github.com/TrustChainEG/postal-codes-json-xml-csv

2. State of the regions, extracted on May 2019, according to
   https://www.deutschepost.de/de/d/deutsche-post-direkt/postleitzahlenkarte.html

3. Daily station observations of mean temperature at 2 m above ground in °C is given by
   https://cdc.dwd.de/portal/. We calculate weekly averages from daily observations and extracted on May 2019.

4. School holidays, extracted on May 2019, according to
   https://www.kmk.org

5. Number of doctors per group listed in the yellow pages from automated web requests searching for 'Arzt' (physician), extracted on May 2019, to
   https://web2.cylex.de

6. Number of hospitals, extracted on May 2019, according to
   http://www.gbe-bund.de

7. Purchasing power per group is transcribed from a map displaying the buying power per district by the GfK Geo-Marketing GmbH, extracted on May 2019,
   http://www.gfk-geomarketing.de

8. Google trends data, extracted on June 2019, is given by https://trends.google.com/trends/

First of all, we map the regions into their states by utilizing https://www.deutschepost.de/de/d/deutsche-post-direkt/postleitzahlenkarte.html. Then, if we have external data, which is only available in the state level like Google trends, we use the state data as the regional data. As a result, the regions, which belong to the same state as obtained by https://www.deutschepost.de/de/d/deutsche-post-direkt/postleitzahlenkarte.html, have a same variable. Besides, we obtain Google trends data via the keyword search of some symptoms since our data includes the sales of some medicinal products. In particular, for each product we use the product specific medical condition as a keyword. For instance, we get the data for the German word 'Erkältung', which is cold in English, and use it as a covariate in the sales prediction of a product, which is consumed for symptomatic relief of cold. Moreover, Google trends data shows search interest relative to the highest point in the data for the given region and time. A value of 100 is the peak popularity for the term and a value of 50 means that the term is half as popular. An example can be seen in Figure 3.2.1.

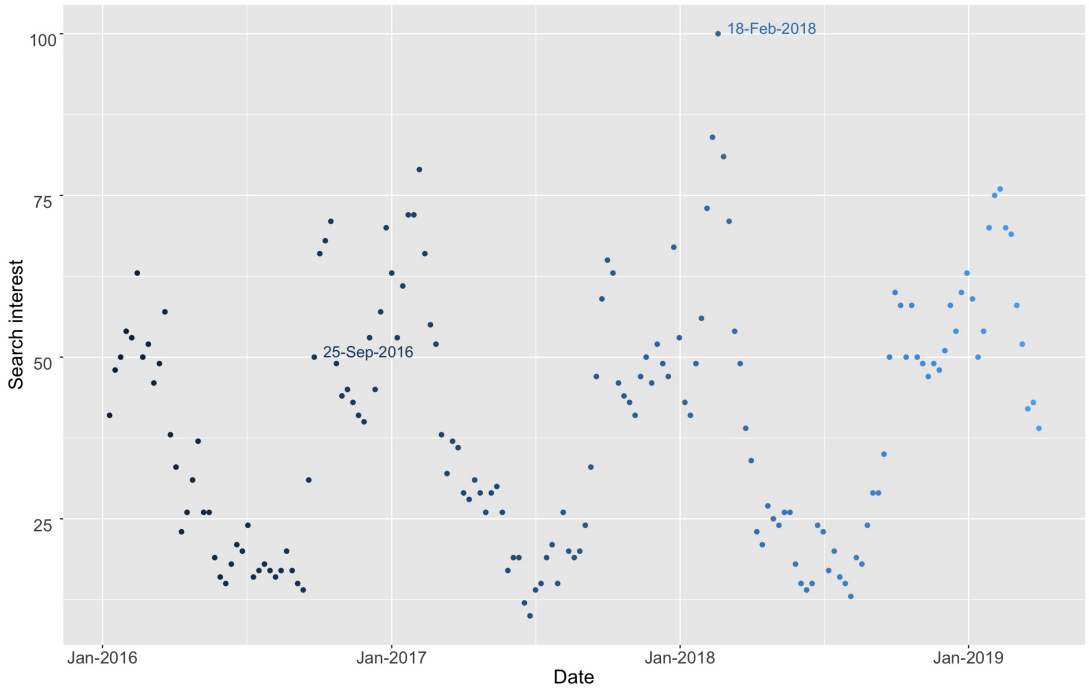


Figure 3.2.1: Weekly Google search interest for the keyword 'Erkältung' in Baden-Würtemberg for the timeline Jan, 10th 2016 and Mar, 31st 2019. Google trends data shows us that the peak for this search term occurred in the week of Feb, 18th 2018 within our time frame. On the other hand, the search in the week of Sep, 25th 2016 were half of the search in the week of Feb, 18th 2018.

3.3 Representative selection

As can be seen in the variable description in Table 3.1.1, we have 95 different `pl` regions. Hence, we later analyze the sales in each `pl` region separately and fit 95 models in Section 4.1. However, we choose some representative regions to compare model performances and to draw conclusions. Our selection depends on the total number of stores in the region j since this covariate has a big effect on the sales of the first product. Moreover, the highest number of the total stores per `pl` region is 59, whereas the lowest number is 5. Thus, we report the results in Section 4.1 with our representative regions given in Table 3.3.1.

Representative regions	
Region	Total number of stores
pl:12	32
pl:38	40
pl:41	53
pl:75	10
pl:82	21

Table 3.3.1: Representatives of all 95 `pl` regions with their total number of stores.

Thus our representative regions for the response variable is given by $representativeregions = \{y_{12tp}, y_{t38p}, y_{t41p}, y_{t75p}, y_{t82p}\}$.

3.4 Data preprocessing steps

We spelled out data preprocessing steps and its importance in Chapter 2. Now we clarify how we follow these steps and what kind results we obtain.

First of all, we identify outliers of $y_{tj_0p_0}$ for $p_0 \in [1, 10]$, $j_0 \in [1, 95]$ and $t = 1, ..., 169$ through LOESS curve fitting and replace them through linearly interpolated values by the `R` package `forecast` implemented by Hyndman et al. (2019). Thus, we analyze $j \cdot p = 95 \cdot 10 = 950$ different time series. In total, the highest number of outliers found in a time series is 22 and the corresponding region has the highest number of the pharmacies. Furthermore, the corresponding medicinal product is the most sold one. Therefore, we conclude that more stores and additionally more sales result in more errors. Moreover, there are some time series, where no outliers are detected. Before we analyze the cleaned data, we approve our decision with the data provider. The company states that since the data comes from the pharmacy software, it is highly likely that a number of pharmacist can be using it in the wrong way thereby leading to extreme values in the data set. Therefore, outlier removal and replacement are justified.

After the data cleaning, we continue with the log-transformation that we introduced in Chapter 2. Since $y_{tjp} \in \mathbb{R}_+$ holds for all p, j and t, we define our response as $log(y_{tjp})$ so

that its prediction is always positive.

Finally, we split the data into the training and test set as discussed in Chapter 2. Hence, $y_{train} = y[1:119]$ and $y_{test} = y[120:169]$ holds, i.e. the data from Jan, 10th 2016 to Apr, 15th 2018 is the training set and from Apr, 22nd 2018 to Mar, 31st 2018 is the test set.

As a software tool, we benefit from `Python` libraries `Pandas` and `NumPy` for these steps, which are presented by McKinney (2010) and Oliphant (2006), respectively.

Chapter 4

Marginal model fitting

In this chapter, we fit different harmonic regression models, discussed in Section 2.4, to the training set of our preprocessed data, explained in Section 3.4. The aim of each model is to predict the weekly number of sales of each product in each `pl` region thereby obtaining residuals and then to estimate the distribution of residuals to gain approximately uniformly distributed data that we can then use as a basis for the copula models, discussed in Section 2.7. Fitting marginal distributions first and fitting the copula parameters for fixed margins afterwards is known as inference functions for margins or the IFM method, as discussed in Joe (1997).

In total we fit two different models to $y_{tj_0p_0}$ for $p_0 \in [1, 10]$, $j_0 \in [1, 95]$ and $t = 1, ..., 169$, which are the harmonic regression model with the specified frequencies and with the frequencies determined by the periodogram analysis, which were introduced in Chapter 2. Moreover, each section contains a short description and comparison of the methods, as well as, how we arrive at the uniformly distributed data sets with the final model.

As a software, we utilize mostly `Python` and its libraries `statsmodels`, `scikit-learn` and `SciPy`, whose details are shown by Seabold et al. (2010), Pedregosa et al. (2011) and Jones et al. (2001), respectively. For visualization purposes, we work with both `Python` library, `Matplotlib`, and `R` package, `ggplot2`. They are presented by Hunter (2007) and Wickham (2016).

4.1 Marginal model specification

As we described before, we choose variables by backward selection based on the Wald test with 0.05 significance level and we report the results for our five example regions, which is the $representativeregions = \{y_{t12p}, y_{t38p}, y_{t41p}, y_{t75p}, y_{t82p}\}$.

Product 1, y_{jt1}

The first product we analyze among the 10 products shows seasonal and cyclic patterns. Therefore, we start with the harmonic regression analysis and then evaluate its performance. The overview of the models fitted is given in Table 4.1.1.

Name	Model	Variable selection criterion
harm.precise	Harmonic regression with specified frequencies	Backward selection via Wald test
harm.spectrum	Harmonic regression with frequencies determined by periodogram analysis	Backward selection via Wald test

Table 4.1.1: Overview of statistical models investigated for Product 1.

Model 1: Harmonic regression model with specified frequencies

We start with the analysis of harmonic regression model that we discussed in Section 2.4. Thus, consider the model for $\ln(y_{tj_01})$ for $j_0 \in [1, 95]$ and $t = 1, ..., 169$, i.e.

$$\begin{aligned}\ln(y_{tj_01}) = intercept + \sum_{k=0}^{3}[\alpha_k \cdot \cos(\omega_k t) + \beta_k \cdot \sin(\omega_k t)] + \beta_{hol} \cdot hol_{tj_0} \\ + \beta_{temp} \cdot temp_{tj_0} + \beta_{google} \cdot google_{tj_01} + \epsilon_t \quad \text{for} \quad t = 1, ..., 169 \quad \text{and} \quad j_0 \in [1, 95].\end{aligned} \tag{4.1.1}$$

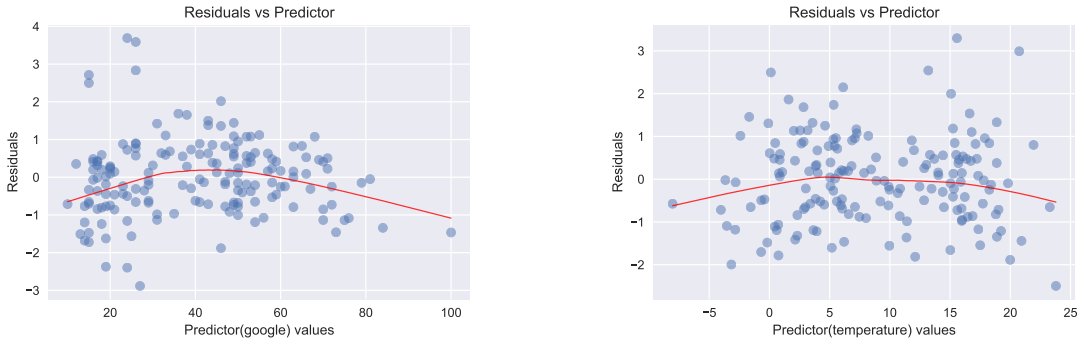
First of all, we analyze the yearly, half-yearly, quarterly, and monthly effects thereby assigning $\omega_1 = \frac{2\pi}{52}$, $\omega_2 = \frac{2\pi}{26}$, $\omega_3 = \frac{2\pi}{13}$ and $\omega_4 = \frac{2\pi}{4}$, respectively. Thus, one model out of 95 harmonic regression models can be more explicitly written as:

$$\begin{aligned}\ln(y_{tj_01}) = intercept + \alpha_0 \cdot \cos(\omega_0 t) + \beta_0 \cdot \sin(\omega_0 t) + \alpha_1 \cdot \cos(\omega_1 t) + \beta_1 \cdot \sin(\omega_1 t) \\ + \alpha_2 \cdot \cos(\omega_2 t) + \beta_2 \cdot \sin(\omega_2 t) + \alpha_3 \cdot \cos(\omega_3 t) + \beta_3 \cdot \sin(\omega_3 t) + \beta_{hol} \cdot hol_{tj_0} \\ + \beta_{temp} \cdot temp_{tj_0} + \beta_{google} \cdot google_{tj_01} + \epsilon_t \quad \text{for} \quad t = 1, ..., 169 \quad \text{and} \quad j_0 \in [1, 95].\end{aligned} \tag{4.1.2}$$

After the backward selection applied to Equation (4.1.2), we check the model diagnostics and realize that the residuals versus google trends plot shows nonlinear curve for most of the `pl` regions, but the residuals versus temperature plot shows a straight line around 0. An example from two out of the five example regions can be seen in Figure 4.1.1. Therefore, we apply log-transformation to the Google trends data and refit the model. Now, our model equation reads:

$$\begin{aligned}\ln(y_{tj_01}) = intercept + \alpha_0 \cdot \cos(\omega_0 t) + \beta_0 \cdot \sin(\omega_0 t) + \alpha_1 \cdot \cos(\omega_1 t) + \beta_1 \cdot \sin(\omega_1 t) \\ + \alpha_2 \cdot \cos(\omega_2 t) + \beta_2 \cdot \sin(\omega_2 t) + \alpha_3 \cdot \cos(\omega_3 t) + \beta_3 \cdot \sin(\omega_3 t) + \beta_{hol} \cdot hol_{tj_0} \\ + \beta_{temp} \cdot temp_{tj_0} + \beta_{google} \cdot \ln(google_{tj_01}) + \epsilon_t \quad \text{for} \quad t = 1, ..., 169 \quad \text{and} \quad j_0 \in [1, 95].\end{aligned} \tag{4.1.3}$$

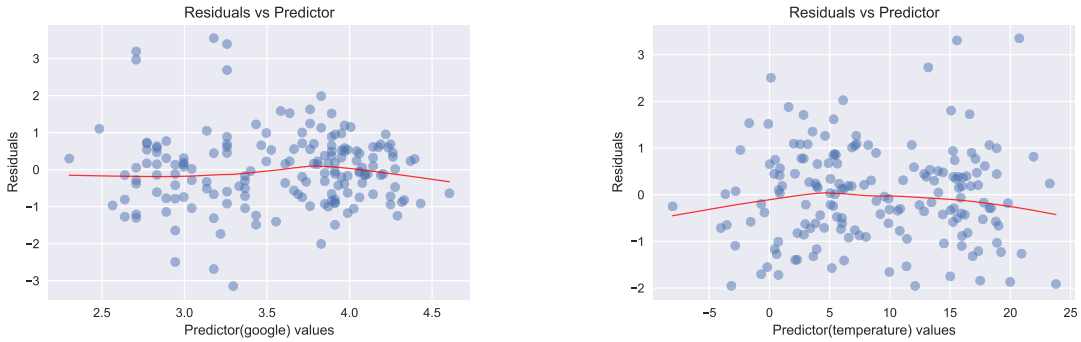
Thus, after the backward selection applied to Equation (4.1.3), we analyze again the model diagnostics. First of all, there are not any patterns in the residuals versus predictor plots and the example of the two regions can be seen in Figure 4.1.2. Secondly, despite of the some small deviations from the optimal cases, the residual versus fitted values plot, normal probability plot of residuals and scale-location plot usually fulfill the assumptions of the harmonic regression model like linearity, equal variance and normality. For instance,



(a) **Region 75** (b) **Region 38**

Figure 4.1.1: Residuals versus predictor plots without transforming the predictor variables. The left plot shows that a nonlinear transformation of the Google trends data is required. The right plot shows that a transformation of the temperature data is not needed.

if we look at the model diagnostics for the region 77 and 61 in Figure 4.1.3 and 4.1.4, respectively, we see that the assumptions of linearity and equal variances are met; although there can be some heteroscedasticity in the model of the region 61. Besides, there are a few isolated points in normal q-q plot, which we assume that they do not violate the normality assumption.



(a) **Region 75** (b) **Region 38**

Figure 4.1.2: Residuals versus predictor plots after nonlinear transformation of the Google trends data. The left plot shows that a nonlinear transformation of the Google trends data is justified. The right plot shows that the temperature data satisfies still the linearity assumption.

Furthermore, the region 77 has the highest R^2_{adj} being equal to 0.99 and the region 61 has the lowest R^2_{adj} being equal to 0.57. The mean value of R^2_{adj} for all 95 regions is 0.80. The boxplots of R^2_{adj} values of harm.precise model is given in Figure 4.2.1. We see that the regions 61, 77 and 82 are the outliers and the median value for R^2_{adj} is above 0.8.

We give the fitted model equations of the reduced models for our representative regions

in Table 4.1.2.

Region	Fitted model equations
pl:12	$\ln(y_{t,12,1}) = 6.01 + 0.25 \cdot \cos(\omega_0 t) - 0.20 \cdot \sin(\omega_0 t) + 0.11 \cdot \sin(\omega_1 t) + 0.11 \cdot \cos(\omega_2 t) - 0.05 \cdot temp_{t,12}$
pl:38	$\ln(y_{t,38,1}) = 3.94 + 0.2 \cdot \cos(\omega_0 t) - 0.14 \cdot \sin(\omega_0 t) - 0.12 \cdot \cos(\omega_1 t) + 0.04 \cdot hol_{t,38} - 0.04 \cdot temp_{t,38} + 0.51 \cdot \ln(google_{t,38,1})$
pl:41	$\ln(y_{t,41,1}) = 3.46 - 0.03 \cdot temp_{t,41} + 0.83 \cdot \ln(google_{t,41,1})$
pl:75	$\ln(y_{t,75,1}) = 1.57 + 0.27 \cdot \cos(\omega_0 t) + 0.82 \cdot \ln(google_{t,75,1})$
pl:82	$\ln(y_{t,82,1}) = 1.35 - 0.02 \cdot temp_{t,82} + 0.91 \cdot \ln(google_{t,82,1})$

Table 4.1.2: Reduced model equations for each of the representative regions' harmonic regression models with specified frequencies after applying sequential backward selection based on the Wald test with 0.05 significance level.

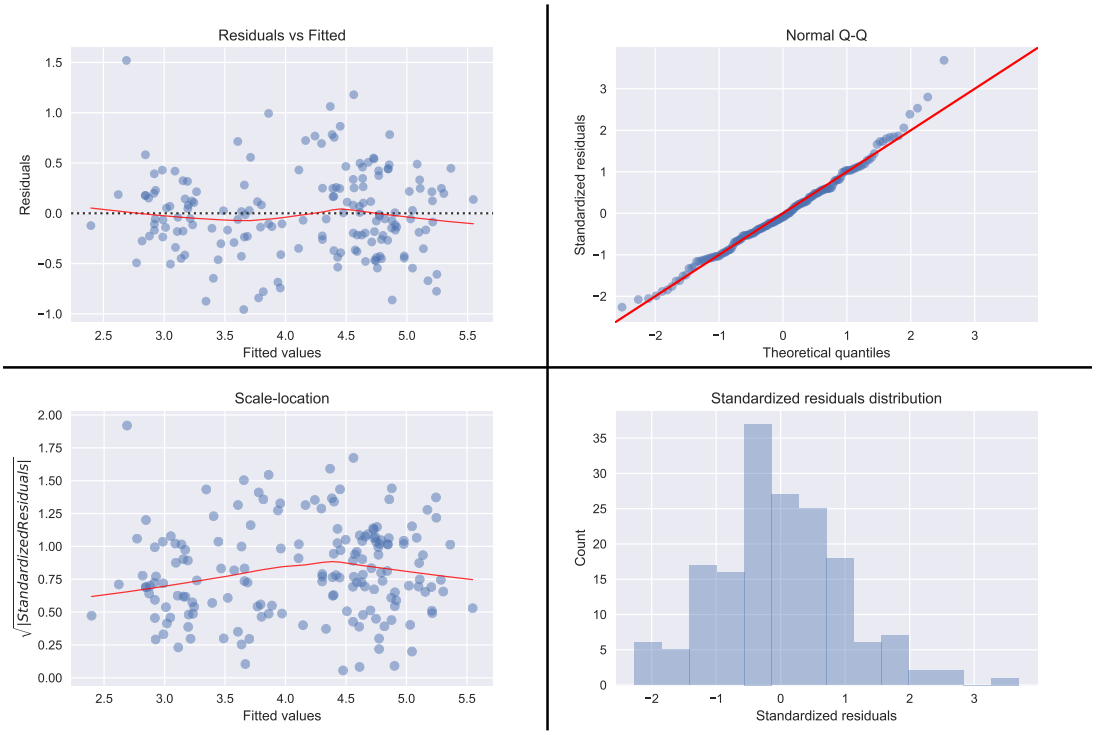


Figure 4.1.3: Regression diagnostics of the harm.precise model of the **region 77**, whose $R^2_{adj} = 0.99$. The plots show that the model assumptions are satisfied, thus we specify our model correctly.

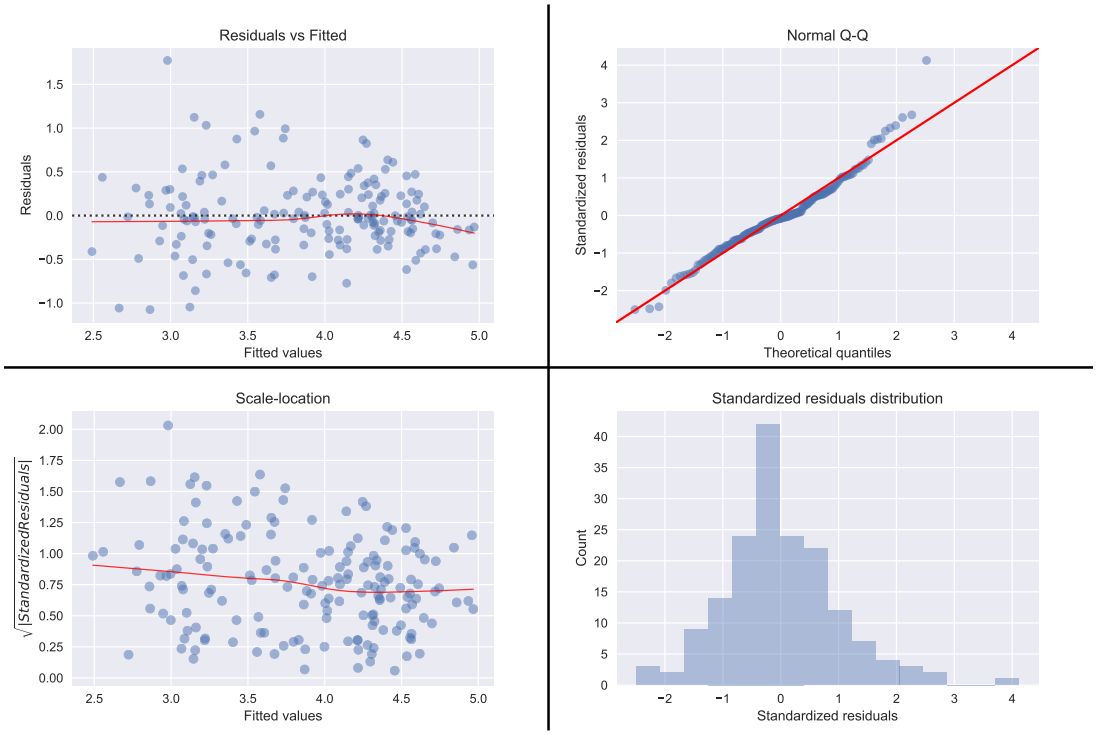


Figure 4.1.4: Regression diagnostics of the harm.precise model of the **region 61**, whose $R^2_{adj} = 0.57$. The plots show that the model assumptions are satisfied, thus we specify our model correctly.

Model 2: Harmonic regression model with frequencies determined by periodogram analysis

In the second model setup, we consider the model given in Equation 4.1.3. Nevertheless, now let ω_k of the given data to be estimated by the associated periodogram. The estimated frequencies for our five example regions are given in Table 4.1.3. From Table 4.1.3, we

Region	ω_0	ω_1	ω_2	ω_3
pl:12	$\frac{2\pi}{60}$	$\frac{2\pi}{40}$	$\frac{2\pi}{17}$	$\frac{2\pi}{12}$
pl:38	$\frac{2\pi}{60}$	$\frac{2\pi}{40}$	$\frac{2\pi}{12}$	$\frac{2\pi}{8}$
pl:41	$\frac{2\pi}{60}$	$\frac{2\pi}{40}$	$\frac{2\pi}{17}$	$\frac{2\pi}{11}$
pl:75	$\frac{2\pi}{60}$	$\frac{2\pi}{40}$	$\frac{2\pi}{11}$	$\frac{2\pi}{10}$
pl:82	$\frac{2\pi}{60}$	$\frac{2\pi}{40}$	$\frac{2\pi}{17}$	$\frac{2\pi}{11}$

Table 4.1.3: Dominant frequency values for each of the representative regions. The frequencies are obtained by periodogram analysis.

can say that the weekly number of sales of the first product in region 12 show periodical patterns in every 3, 4, 10 and 15 months. Similarly, the number of the sales of the first product in region 38 show periodical patterns in every 2, 3, 10 and 15 months. In the harmonic regression model, now we analyze the 4 frequency values, which have a higher spectrum than others, as follows:

$$\ln(y_{tj_01}) = intercept + \sum_{k=0}^{3}[\alpha_k \cdot \cos(\omega_k t) + \beta_k \cdot \sin(\omega_k t)] + \beta_{hol} \cdot hol_{tj_0} + \beta_{temp} \cdot temp_{tj_0} + \beta_{google} \cdot \ln(google_{tj_01}) + \epsilon_t \quad \text{for} \quad t = 1, ..., 169 \quad \text{and} \quad j_0 \in [1, 95]. \tag{4.1.4}$$

After the backward selection and regression diagnostic check, the fitted model equations of the reduced models for our representative regions are given in Table 4.1.4. Besides, the

Region	Fitted model equations
pl:12	$\ln(y_{t,12,1}) = 4.26 - 0.05 \cdot \sin(\omega_0 t) + 0.06 \cdot \sin(\omega_1 t) - 0.10 \cdot \cos(\omega_2 t) - 0.03 \cdot \cos(\omega_3 t) - 0.05 \cdot temp_{t,12} + 0.45 \cdot \ln(google_{t,12,1})$
pl:38	$\ln(y_{t,38,1}) = 4.35 - 0.23 \cdot \sin(\omega_0 t) + 0.11 \cdot \cos(\omega_0 t) + 0.26 \cdot \sin(\omega_1 t) + 0.08 \cdot \sin(\omega_3 t) - 0.03 \cdot temp_{38,t} + 0.41 \cdot \ln(google_{38,t,1})$
pl:41	$\ln(y_{t,41,1}) = 3.99 + 0.08 \cdot \sin(\omega_1 t) - 0.10 \cdot \cos(\omega_2 t) - 0.08 \cdot \cos(\omega_3 t) - 0.04 \cdot temp_{t,41} + 0.81 \cdot \ln(google_{t,41,1})$
pl:75	$\ln(y_{t,75,1}) = 1.96 + 0.10 \cdot \sin(\omega_1 t) - 0.15 \cdot cos(\omega_2 t) + 0.15 \cdot \sin(\omega_3 t) - 0.03 \cdot temp_{t,75} + 0.80 \cdot \ln(google_{t,75,1})$
pl:82	$\ln(y_{t,82,1}) = 1.46 + 0.11 \cdot \sin(\omega_1 t) - 0.07 \cdot cos(\omega_3 t) - 0.02 \cdot temp_{t,82} + 0.90 \cdot \ln(google_{t,82,1})$

Table 4.1.4: Reduced fitted model equations for each of the representative regions' harmonic regression models, whose frequencies are obtained by periodogram analysis, after applying sequential backward selection based on the Wald test with 0.05 significance level.

region 77 has the highest R^2_{adj} being equal to 0.99 and the region 61 has the lowest R^2_{adj} being equal to 0.57 like the result of first model. The mean value of R^2_{adj} for all 95 regions is 0.81.

4.2 Marginal models comparison

In the previous part, we specified the two marginal models and introduced their fitting results. In this section, we compare these models with each other. More specifically, we analyze their performance and significant variables for our five representative regions.

First of all, the harm.spectrum model is not significantly better than the harm.precise model as can be seen in the boxplot given in Figure 4.2.1 since the median R^2_{adj} value for the models is almost same, above 0.80. Moreover, we see that the outlier regions, which are `pl` **61, 77** and **88**, are the same for the both harm.precise and harm.spectrum model.

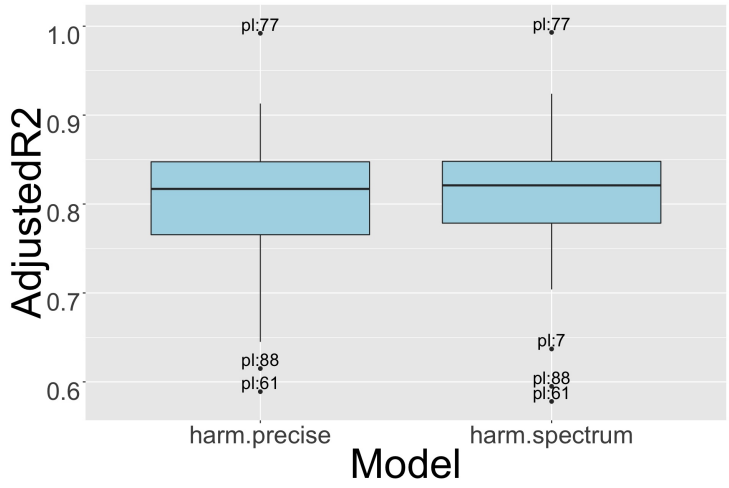


Figure 4.2.1: In sample comparison: Boxplots of R^2_{adj} values for the harm.precise and the harm.spectrum model fitted to all 95 `pl` regions.

Furthermore, we do not observe any significant difference in the distribution of R^2_{adj} values in Figure 4.2.2.

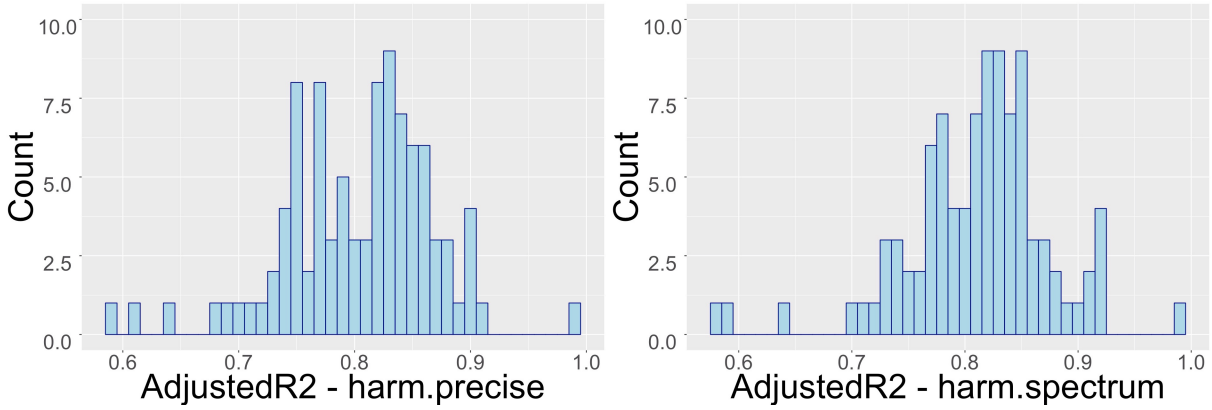


Figure 4.2.2: In sample comparison: Histogram of R^2_{adj} values for the harm.precise and the harm.spectrum model fitted to all 95 `pl` regions.

Secondly, Table 4.2.1 summarizes the results, including the RMSPE, introduced in Section 2.5, for the training and test set. We can see that the model harm.precise always has the smaller RMSPE on the test set than the model harm.spectrum. On the other hand, the comparison is opposite for the training set. Thus, we conclude that the model harm.spectrum is overfitting.

Moreover, the model diagnostic of harm.precise is acceptable as shown before in Figure 4.1.3 and 4.1.4. Thus, we continue our analysis with the adjusted R-squared values of the model harm.precise in Section 5 and with the residuals of the model harm.precise in Section 7.

Model	`pl` region	R^2_{adj}	$RMSPE_{train}$	$RMSPE_{test}$	Significant variables
harm.precise	**pl:12**	0.77	86.13	67.40	$\cos(\omega_0)$, $\sin(\omega_0)$, $\sin(\omega_1)$, $\cos(\omega_2)$, temp
harm.spectrum	**pl:12**	0.80	80.74	123.31	$\sin(\omega_0)$, $\sin(\omega_1)$, $\cos(\omega_2)$, $\cos(\omega_3)$, temp, google
harm.precise	**pl:38**	0.82	104.06	132.97	$\cos(\omega_0)$, $\sin(\omega_0)$, $\cos(\omega_1)$, temp, hol, google
harm.spectrum	**pl:38**	0.84	94.57	151.15	$\sin(\omega_0)$, $\cos(\omega_0)$, $\sin(\omega_1)$, $sin(\omega_3)$, temp, google
harm.precise	**pl:41**	0.84	131.26	125.52	temp, google
harm.spectrum	**pl:41**	0.85	114.10	158.40	$\sin(\omega_1)$, $\cos(\omega_2)$, $\cos(\omega_3)$, temp, google
harm.precise	**pl:75**	0.78	37.11	39.81	$\cos(\omega_0)$, google
harm.spectrum	**pl:75**	0.83	34.59	44.00	$\sin(\omega_1)$, $\cos(\omega_2)$, $\sin(\omega_3)$, temp, google
harm.precise	**pl:82**	0.83	39.11	42.57	temp, google
harm.spectrum	**pl:82**	0.85	36.72	43.41	$\sin(\omega_1)$, $\cos(\omega_3)$, temp, google

Table 4.2.1: Results of the different harmonic regression models for the weekly number of the sales of the first product in the **Region 12, 38, 41, 75** and **82**. The best model for each region is highlighted in grey.

Chapter 5

Exploration of spatial dependence patterns

In the previous sections, we obtained our final harmonic regression model and calculated R^2_{adj} for each `pl` region. For this aim, we did not use our time independent variables, which were introduced in Table 3.1.1. In this chapter, we analyze whether there is either linear or nonlinear dependence between our model power, R^2_{adj}, and the time independent variables.

For this aim, we first of all scale our time independent variables and R^2_{adj} values to zero mean and unit variance. Next, we investigate the value difference between each variable and R^2_{adj}. We present the visualization of it on the Germany map in Figure 5.0.1 using `Tableau`.

In Figure 5.0.1, the white color corresponds to the value zero, which represents the equal values of R^2_{adj} and the corresponding variable. Therefore, what the plot shows us, is that there is not any exact linear dependence between the corresponding variable and our model power, R^2_{adj} since the dominant color on the map is not white. What we additionally analyze, is the nonlinear dependence between our model power and the time independent variables. Thus, we investigate the relationship between each variable and R^2_{adj} values with the LOESS curve, introduced in Section 3.4. All plots in Figure 5.0.2 shows that there exists the nonlinear dependence between the time independent variables and R^2_{adj} since the fitted curves are nonlinear. However, we remind that our conclusion results from a subset of pharmacies, which uses the our data provider's software. Hence, it may not reflect the general overview for Germany.

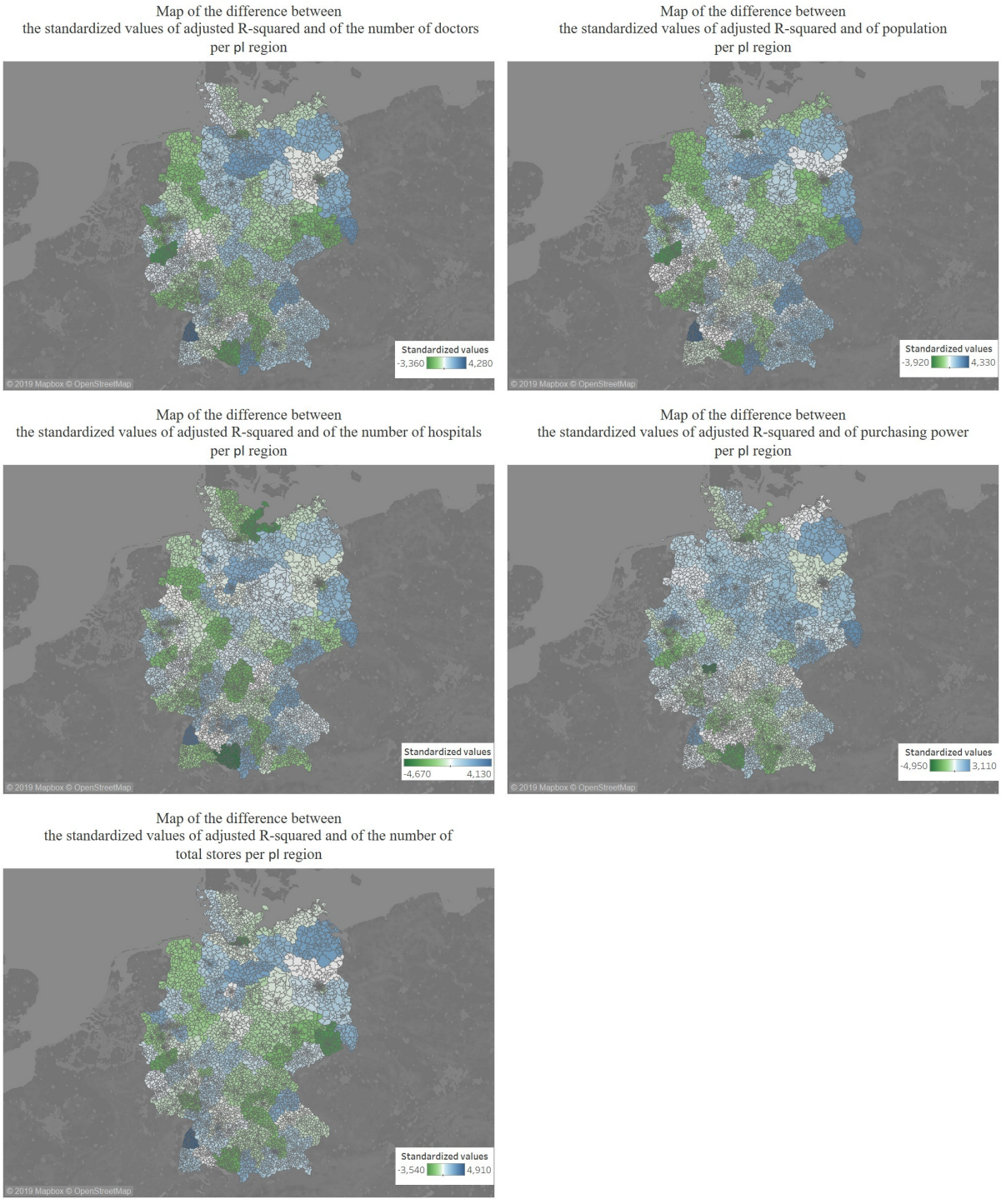


Figure 5.0.1: The distribution of the difference between R^2_{adj} and population, the number of total stores, doctors, hospitals, purchasing power over all Germany per `pl` region. The values for each variable and R^2_{adj} are standardized so that the mean is zero and the variance is 1.

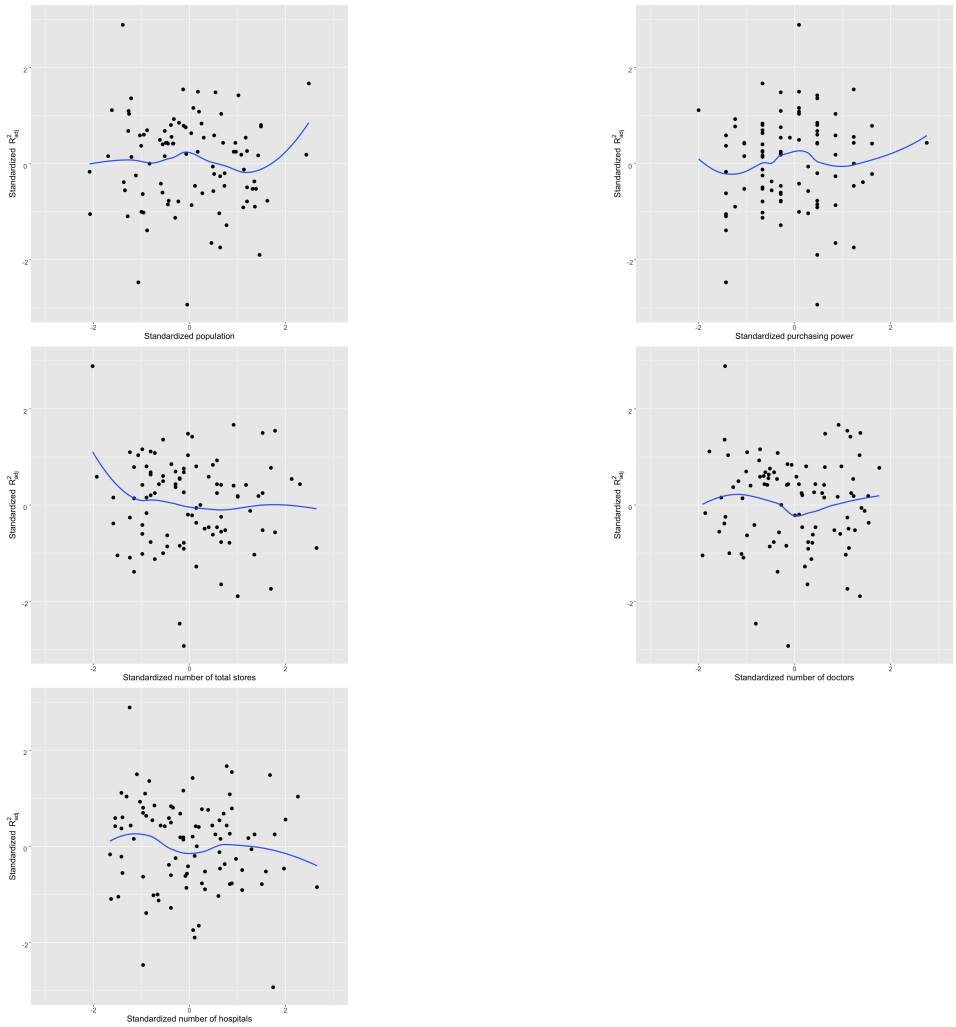


Figure 5.0.2: Scatter plots of the standardized time independent variables and the standardized R^2_{adj} values per `pl` region. The blue smoothing curve is obtained by the LOESS curve fitting approach.

Chapter 6

Clustering

In Section 2.6, we discussed and compared some methods to obtain clusterings and determine the number of clusters. In this chapter, we test these methods by applying them to an actual time series data set. We do this in R, where we implement a selection of methods and compare them based on the performance. We follow the feature-based time series clustering approach that we discussed in Section 2.6.

Even though it is often hard to measure the performance of unsupervised learning algorithms like clustering algorithms since what makes a clustering 'good' is subjective, our ultimate goal is to find the clusters, whose regions show similar sales behavior. If we can obtain such clusters, we analyze the dependencies within them in Section 7.

In the upcoming sections, we explain the setup of our analysis and present our findings. We start with Section 6.1, where we discuss the time series data set we use for the clustering analysis. Here we additionally discuss the features we selected. Next, in Section 6.2, we discuss the implementations and apply the three indices that we discussed in Section 2.6.3 to determine the number of clusters in which the data should be partitioned. Moreover, we present our final results on the performance of the different algorithms to obtain clusterings and compare them with each other again in Section 6.2. Last, we explore the clusters of the best clustering in Section 6.2.4.

6.1 Feature selection

In this section we discuss the time series data set and the features that we use in our practical research. Because we aim to investigate the sales behavior of the different `pl` regions and to cluster them accordingly, we start with the analysis of the predictions of the time dependent variables, which were given in Table 3.1.1. More specifically, we

rewrite our harm.precise model, which was discussed in Section 4.1, as:

$$\ln(y_{tj_01}) = intercept + \beta_{seasonal} \cdot seasonal_{tj_0} + \beta_{temp} \cdot temp_{tj_0} + \beta_{google} \cdot \ln(google_{tj_01}) + \epsilon_t \quad \text{for} \quad t = 1, ..., 169 \quad \text{and} \quad j_0 \in [1, 95], \tag{6.1.1}$$

where $\beta_{seasonal} \cdot seasonal_{tj_0}$ is defined as follows:

$$\beta_{seasonal} \cdot seasonal_{tj_0} := \sum_{k=0}^{3} [\alpha_k \cdot \cos(\omega_k t) + \beta_k \cdot \sin(\omega_k t)] + \beta_{hol} \cdot hol_{tj_0} \quad \text{for} \quad t = 1, ..., 169 \quad \text{and} \quad j_0 \in [1, 95]. \tag{6.1.2}$$

Moreover, we define the new variables $r_{tj_0}^{google}$, $r_{tj_0}^{temp}$, and $r_{tj_0}^{seasonal}$, which correspond to the prediction percentage of the variables $\ln(google_{tj_01})$, $temp_{tj_0}$ and $seasonal_{tj_0}$, respectively, as follows:

$$p_{tj_0}^{google} := \hat{\beta}_{google} \cdot \ln(google_{tj_01}) \quad \text{for} \quad t = 1, ..., 169 \quad \text{and} \quad j_0 \in [1, 95], \tag{6.1.3}$$

$$p_{tj_0}^{temp} := \hat{\beta}_{temp} \cdot temp_{tj_0} \quad \text{for} \quad t = 1, ..., 169 \quad \text{and} \quad j_0 \in [1, 95], \tag{6.1.4}$$

$$p_{tj_0}^{seasonal} := \hat{\beta}_{seasonal} \cdot seasonal_{tj_0} \quad \text{for} \quad t = 1, ..., 169 \quad \text{and} \quad j_0 \in [1, 95], \tag{6.1.5}$$

$$r_{tj_0}^{google} := \frac{|p_{tj_0}^{google}|}{|p_{tj_0}^{google}| + |p_{tj_0}^{temp}| + |p_{tj_0}^{seasonal}|} \quad \text{for} \quad t = 1, ..., 169 \quad \text{and} \quad j_0 \in [1, 95], \tag{6.1.6}$$

$$r_{tj_0}^{temp} := \frac{|p_{tj_0}^{temp}|}{|p_{tj_0}^{google}| + |p_{tj_0}^{temp}| + |p_{tj_0}^{seasonal}|} \quad \text{for} \quad t = 1, ..., 169 \quad \text{and} \quad j_0 \in [1, 95], \tag{6.1.7}$$

$$r_{tj_0}^{seasonal} := \frac{|p_{tj_0}^{seasonal}|}{|p_{tj_0}^{google}| + |p_{tj_0}^{temp}| + |p_{tj_0}^{seasonal}|} \quad \text{for} \quad t = 1, ..., 169 \quad \text{and} \quad j_0 \in [1, 95]. \tag{6.1.8}$$

As can be seen above, our new data contains weekly time series data on three variables: the prediction percentage of the google trends, seasonal and temperature components. We plot our time series data given in Equations (6.1.6), (6.1.7) and (6.1.8) in Figure 6.1.1. We see that the prediction percentage of the Google trends term is less in the summer, whereas, the prediction percentage of the temperature term is higher in the summer. Moreover, the prediction percentage of the seasonal term changes region by region. For example, its prediction percentage is higher in the summer for the `pl 52`, but is lower in the summer for the `pl 8`.

As a next step for the setup of the clustering, it is important to decide on which features should be selected from the time series data. As we discussed in Section 2.6, several feature selection methods are available and application dependent. Depending on the goal of the clustering, it may be desired to, for example, use the mean, standard deviation and even entropy of the data. In our case, we decided to extract the mean and standard deviation of the time series data given in Equation (6.1.6), (6.1.7) and (6.1.8), since we want to the capture the average effect of the google trends, temperature and seasonal components on the sales, as well as the amount of variation of them. These selected features are given

by the formulas:

$$s_{j_0}^{google} := \hat{\mu}_{t=1,\dots,169}(r_{j_0t}^{google}) \quad \text{for} \quad j_0 \in [1, 95], \tag{6.1.9}$$

$$s_{j_0}^{temp} := \hat{\mu}_{t=1,\dots,169}(r_{j_0t}^{temp}) \quad \text{for} \quad j_0 \in [1, 95], \tag{6.1.10}$$

$$s_{j_0}^{seasonal} := \hat{\mu}_{t=1,\dots,169}(r_{j_0t}^{seasonal}) \quad \text{for} \quad j_0 \in [1, 95], \tag{6.1.11}$$

$$z_{j_0}^{google} := \hat{\sigma}_{t=1,\dots,169}(r_{j_0t}^{google}) \quad \text{for} \quad j_0 \in [1, 95], \tag{6.1.12}$$

$$z_{j_0}^{temp} := \hat{\sigma}_{t=1,\dots,169}(r_{j_0t}^{temp}) \quad \text{for} \quad j_0 \in [1, 95], \tag{6.1.13}$$

$$z_{j_0}^{seasonal} := \hat{\sigma}_{t=1,\dots,169}(r_{j_0t}^{seasonal}) \quad \text{for} \quad j_0 \in [1, 95]. \tag{6.1.14}$$

Before moving on to the actual clustering analysis, it must be decided additionally on which preprocessing should be applied to the data. In our case, we decide to apply normalization for the features, because we want to cluster based on similar shapes. In normalization, the data is scaled into a range of $[0, 1]$ as explained in Section 2.2.2. Therefore, each of the 6 features is scaled to the interval $[0, 1]$.

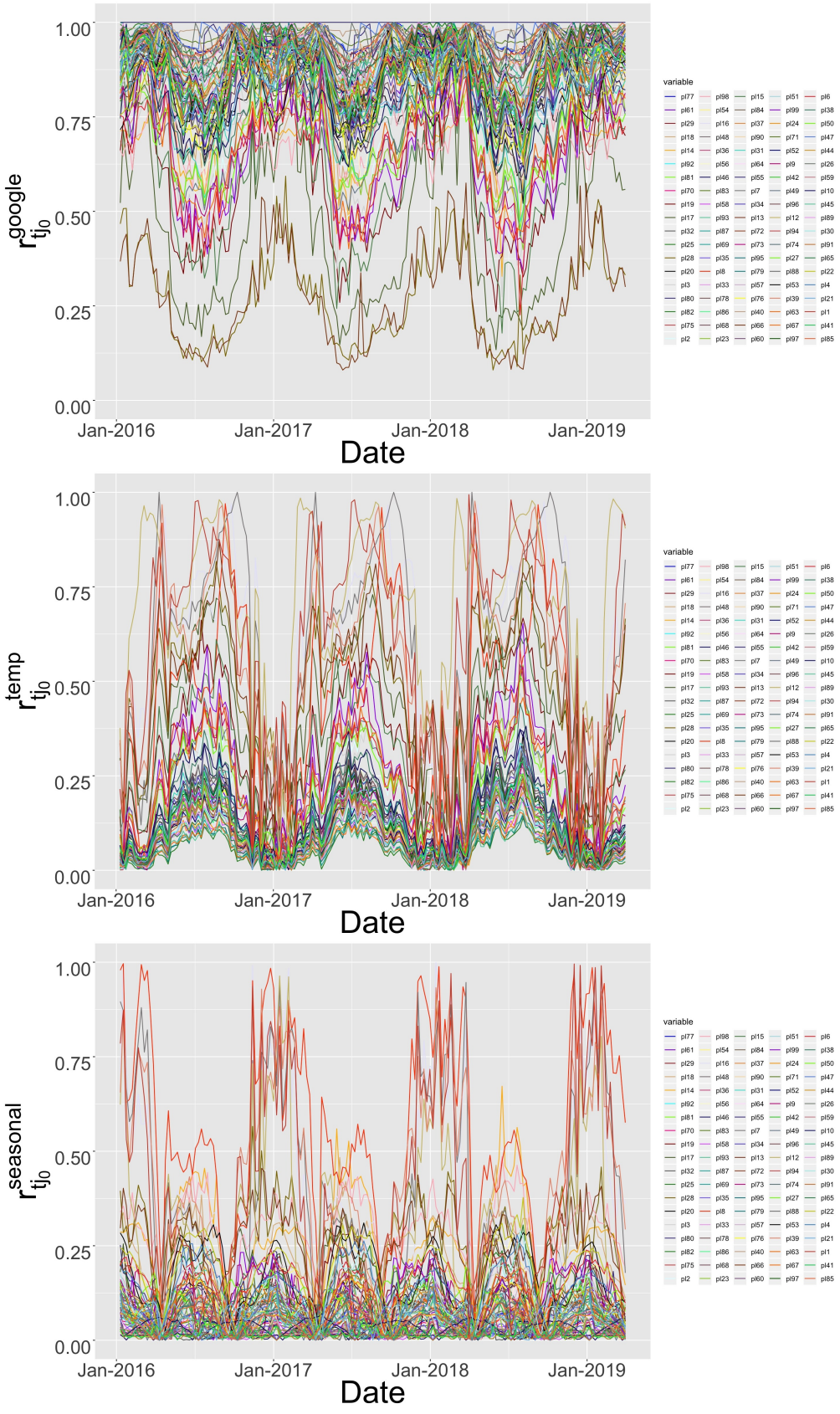


Figure 6.1.1: Plots of the variables defined in Equations (6.1.6), (6.1.7) and (6.1.8). Each curve in the plot corresponds to a `pl` region.

6.2 Distance-based clustering analysis

After we decided our features for obtaining the clusterings, in this section we discuss the R implementations of all validation indices and the distance-based clustering methods that we discussed in Section 2.6.2 and the different setups for the clustering. Moreover, we apply all indices for k-means clustering and agglomerative hierarchical clustering that we discussed in Section 2.6.3 based on the implementations being available in R. For the hierarchical clustering, we apply both average linkage and Ward's method that were introduced in Section 2.6.2.

Before looking into the various clustering setups and their results, below we give an overview of the clustering steps that we follow:

1. Create three time series data sets, which contain the prediction percentage of the predictor variables: google trends, seasonal and temperature.
2. Extract the mean and standard deviation of the time series as features.
3. Normalize the selected features into a range of $[0, 1]$.
4. Use Euclidean distance as a similarity, distance measure.
5. Use k-means clustering and agglomerative hierarchical clustering as the clustering methods.
6. Decide on the clustering validation indices.
7. Decide on the range of the possible number of clusters.
8. Perform clustering with the different set of the features.
9. Find the 'optimal' number of clusters based on the validation score for each clustering method and the feature set.
10. Select the best number of clusters and the clustering method.
11. Analyze the 'optimal' clustering in detail and visualize it.

The first six steps were explained in the previous section and in Section 2.6.2. By visual inspection with Figure 6.1.1, we decided that at least three and at most six clusters should be obtained from the clustering. We therefore set the range of the possible number of clusters from 3 to 6 clusters. For the eighth step, we benefit from the R implementations of the different indices and the clustering methods. Below we give a brief summary of them:

Silhouette index: Implementation available in the *NbClust* function of the `NbClust` package given by Charrad et al. (2014).

Gamma index: Implementation available in the *NbClust* function of the `NbClust` package.

Agglomerative hierarchical clustering: We applied it by using the *hclust* function of the `stats` package given by R Core Team (2018). Because average linkage and Ward's method seem to generate the best clustering, we only consider these two methods.

***k*-means**: We applied it by using the *kmeans* function of the `stats` package with the Hartigan-Wong algorithm.

In the upcoming sections, we perform clustering with the different set of the features.

6.2.1 1st scenario: Mean of the time series data sets as all features

In this experiment, we use the mean of our three time series data sets as features rather than using the mean of each time series data set as only one feature at a time. Therefore, the variables defined in Equations (6.1.9), (6.1.10) and (6.1.11) are the features for obtaining a clustering. Next, we again run agglomerative hierarchical and k-means clustering methods with the range of the number of clusters from 3 to 6 and compare the validation indices.

We give a summary of the results in Table 6.2.1. We begin by observing that the 'optimal' number of clusters of 3 seems to hold since the majority of the indices agree on 3 clusters. Therefore, we conclude that the 'optimal' number of clusters is 3 thanks to the clear agreement between the indices. Furthermore, what we conclude, is that the agglomerative hierarchical clustering method with the average linkage is the best performing method in this case. There is a clear difference between the indices of the agglomerative hierarchical clustering method and the others. Because the higher index shows the better method, the first column of the Table 6.2.1 is our best method, agglomerative hierarchical clustering method with average linkage.

Feature set	Number of clusters	Agglomerative hierarchical (average linkage)		Agglomerative hierarchical (Ward's method)		k-means	
		Silhouette index	Gamma index	Silhouette index	Gamma index	Silhouette index	Gamma index
$s_{j_0}^{google}, s_{j_0}^{temp}, s_{j_0}^{seasonal}$	3	0.71	0.99	0.56	0.90	0.54	0.86
	4	0.71	0.98	0.52	0.89	0.48	0.81
	5	0.54	0.90	0.51	0.88	0.50	0.86
	6	0.55	0.90	0.51	0.89	0.47	0.86

Table 6.2.1: Overview of the clustering validation indices to obtain the range of the number of clusters from 3 to 6 of all combinations of the methods that we applied. The feature set was defined in Equations (6.1.9), (6.1.10) and (6.1.11). The best scores for each method are highlighted in grey. The higher the silhouette index or the gamma index, the better the clustering.

6.2.2 2nd scenario: Mean and standard deviation of time series data sets as all features

As the second setup, both the mean and standard deviation of our time series data sets are used as the features, i.e. the variables defined in Equations (6.1.9), (6.1.10), (6.1.11), (6.1.12), (6.1.13) and (6.1.14) are the features to obtain a clustering. Next, the same runs as in the first scenario are performed.

In Table 6.2.2, the clustering validation indices can be found for each combination of the number of clusters and the clustering method that we considered. The 'optimal' number of cluster for each index and method is highlighted in grey. As can be seen, the same indices and clustering methods from the previous section are applied. We observe that many methods return 3 as the 'optimal' number of clusters; however, no clear agreement between the indices can be detected. Even though 3 is returned by the majority of indices, it does not necessarily mean that this is the optimal number, but rather that no clear clustering structure is present in the data. Moreover, we can still conclude that the k-means clustering method is the best method in this case since its indices agree on 3 clusters and the index values of it are higher than the index values of the other two methods.

Feature set	Number of clusters	Agglomerative hierarchical (average linkage)		Agglomerative hierarchical (Ward's method)		k-means	
		Silhouette index	Gamma index	Silhouette index	Gamma index	Silhouette index	Gamma index
$s_{j_0}^{google}, s_{j_0}^{temp}, s_{j_0}^{seasonal}$ $z_{j_0}^{google}, z_{j_0}^{temp}, z_{j_0}^{seasonal}$	3	0.58	0.90	0.58	0.89	0.58	0.90
	4	0.55	0.90	0.49	0.84	0.49	0.84
	5	0.50	0.92	0.52	0.88	0.36	0.75
	6	0.49	0.91	0.50	0.90	0.44	0.85

Table 6.2.2: Overview of the clustering validation indices to obtain the range of the number of clusters from 3 to 6 of all combinations of the methods that we applied. The feature set was defined in Equations (6.1.9), (6.1.10), (6.1.11), (6.1.12), (6.1.13) and (6.1.14). The best scores for each method are highlighted in grey. The higher the silhouette index or the gamma index, the better the clustering.

6.2.3 Comparison of the scenarios

We run several clustering algorithms and calculated their indices in the previous sections. What we concluded in the 1st scenario, is that applying the agglomerative hierarchical clustering with average linkage to the mean of our all time series data set returns the best clustering of three clusters. In addition, we concluded that selecting features as both the mean and standard deviation of our time series data sets results in unclear clustering structure, but k-means clustering with three clusters seems to be the best option in the last scenario.

In Table 6.2.3, we compare the clustering validation indices of the best option of each scenario as we discussed in the previous section. We observe that both the gamma index and silhouette index of the agglomerative hierarchical clustering with average linkage, where the feature set contains $s_{j_0}^{google}$, $s_{j_0}^{temp}$ and $s_{j_0}^{seasonal}$, is higher than of the other clustering setup. Thus, we select the best clustering based on the explanation power of the clustering results. The method that performs best in our research is the agglomerative hierarchical clustering with average linkage and the features we use are the mean of the time series defined in Equations (6.1.9), (6.1.10), (6.1.11). The more details and results of this scenario can be found in Section 6.2.4.

Clustering method	Feature set	Number of clusters	Silhouette index	Gamma index
Agglomerative hierarchical (average linkage)	$s_{j_0}^{google}, s_{j_0}^{temp}, s_{j_0}^{seasonal}$	3	0.71	0.99
k-means	$s_{j_0}^{google}, s_{j_0}^{temp}, s_{j_0}^{seasonal}, z_{j_0}^{google}, z_{j_0}^{temp}, z_{j_0}^{seasonal}$	3	0.58	0.90

Table 6.2.3: Clustering validation indices of each of the best method given in Section 6.2.1 and Section 6.2.2. The best scores for each method are highlighted in grey. The higher the silhouette index or the gamma index, the better the clustering. The features were defined in Equations (6.1.9), (6.1.10), (6.1.11), (6.1.12), (6.1.13) and (6.1.14).

6.2.4 The exploration of the best clustering, pl region segmentation

After finding the 'optimal' clustering setup, now, we visually analyze the clusters. We note that our 'optimal' clustering setup is to use the mean of our time series data sets and run the agglomerative hierarchical clustering of three clusters with average linkage.

In Figure 6.2.1 the dendrogram, introduced in Section 2.6.2, of the best clustering can be seen. We note that white observations, e.g. pl regions, belong to the second cluster. Similarly, green represents the first cluster and blue corresponds to the third cluster. Moreover, we show a Germany map of the pl regions within each cluster in Figure 6.2.2. It furthermore gives the pl region segmentation for the company.

When visually analyzing the best performing clustering assignment, we observe the differences in the cluster sizes. Our 'optimal' clustering has one big cluster, colored by white, that accounts for 90% of the pl regions and other clusters with only a few (3 and 6) pl regions. For the two small clusters, colored by green and blue, we present additionally a Germany map only with their pl regions in Figure 6.2.3 and Figure 6.2.4. Furthermore, we investigate if any other differences in the obtained clusters are observed. Thus, for each obtained cluster we plot the boxplots of the time independent variables, given in Table 3.1.1, and the boxplots of the mean of our time series data sets.

In Figure 6.2.5, we see that our 'optimal' clustering setup captures the different prediction power of the explanatory variables, introduced in Table 3.1.1. In obtained three clusters, we clearly observe that the second cluster, corresponding to white color in the plots, contains the pl regions whose sales are mostly explained by the Google trends data. Because the second cluster is the biggest cluster, we conclude that the total sales of many pl regions are mostly explained by the Google trends data. Additionally, the second cluster has the pl regions with higher purchasing power. We therefore can infer that as the financial power of a region increases, the chance of understanding the people's behavior from the online services increases. Even though the population, number of doctors and the number of hospitals of the second cluster is additionally higher compared to the other clusters, it can result from the fact that it contains much more pl regions in it. Furthermore, the pl regions without the explanatory power of the Google trends data belongs to the first cluster, colored by green in the plots. The number of the pl regions within this cluster is limited to 6; however, what is interesting, is that the first cluster has the highest median value of the total number of stores, 35.5. Thus, what we can conclude, is that people may not reflect their buying tendency online if there is an enough number of stores to obtain products. Moreover, when comparing the time independent variables of the first and third clusters, corresponding to green and blue colors in the plots, respectively, we do not observe any obvious differences expect the total number of stores and purchasing power. However, the clear difference can be seen for the clustering features. The first cluster consists of the pl regions without the explanatory power of the Google trends data, whereas the pl regions in the third cluster has some predictions made by the Google trends data. We note that although the purchasing power and total number of stores can point out some results, we cannot be conclusive on whether these

are always the case, due to the sales variation in different pharmacy products that we did not analyze.

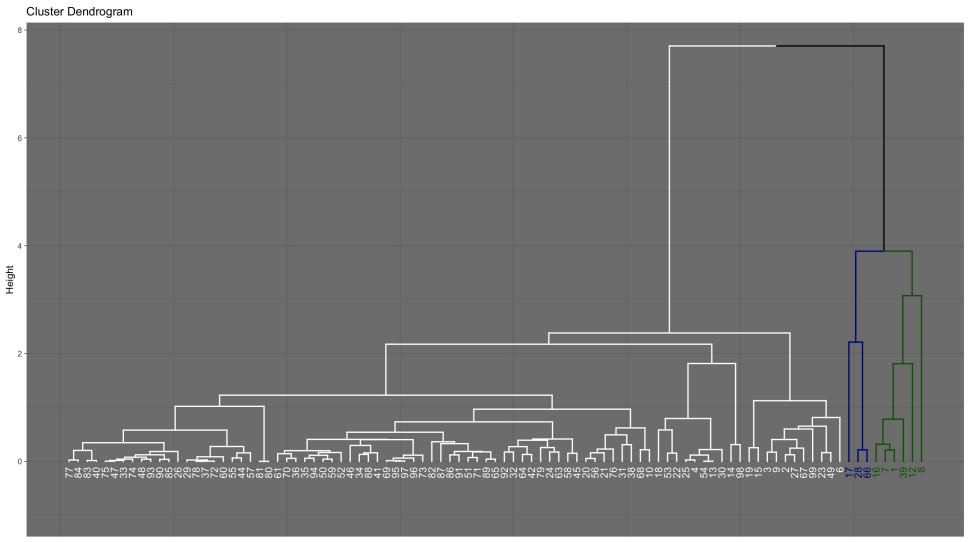


Figure 6.2.1: Dendrogram of the three clusters that are obtained when applying the agglomerative hierarchical clustering with average linkage to the mean of our time series data sets. This clustering assignment is the best in our experiments.

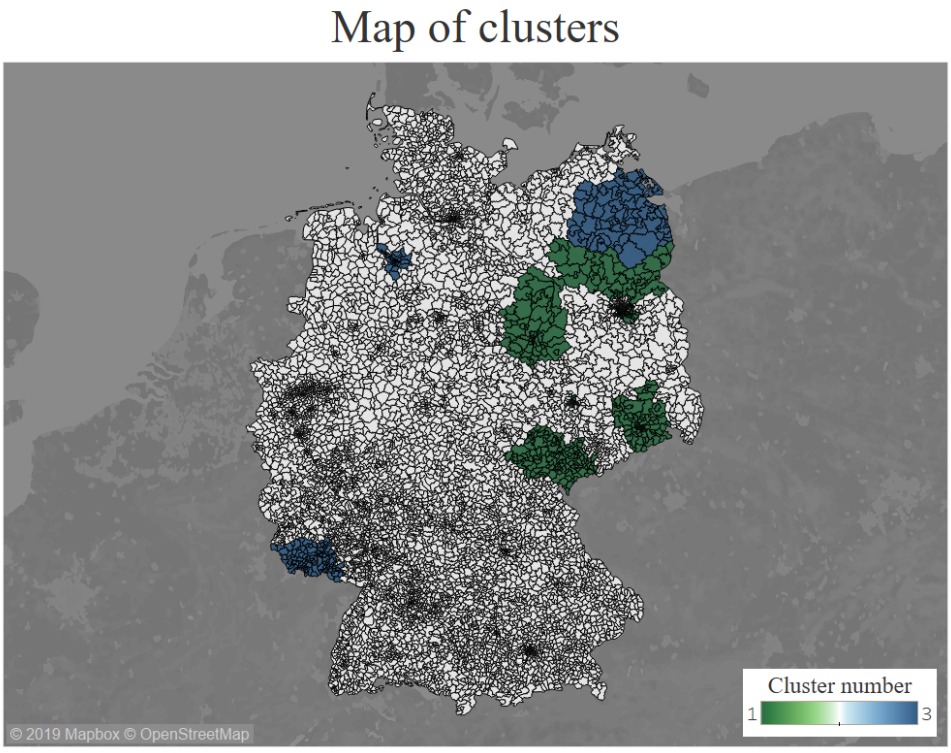


Figure 6.2.2: Visualization of the three clusters that are obtained when applying the agglomerative hierarchical clustering with average linkage to the mean of our time series data sets. This clustering assignment is the best in our experiments.

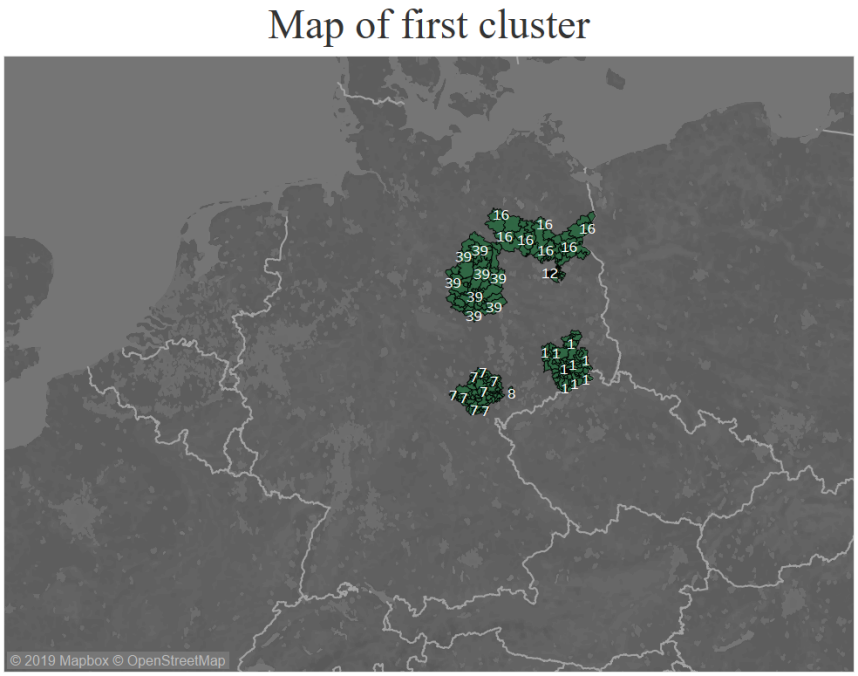


Figure 6.2.3: Visualization of the first cluster that is obtained when applying the agglomerative hierarchical clustering with average linkage to the mean of our time series data sets. It contains `pl` **1**, `pl` **7**, `pl` **8**, `pl` **12**, `pl` **16** and `pl` **39**.

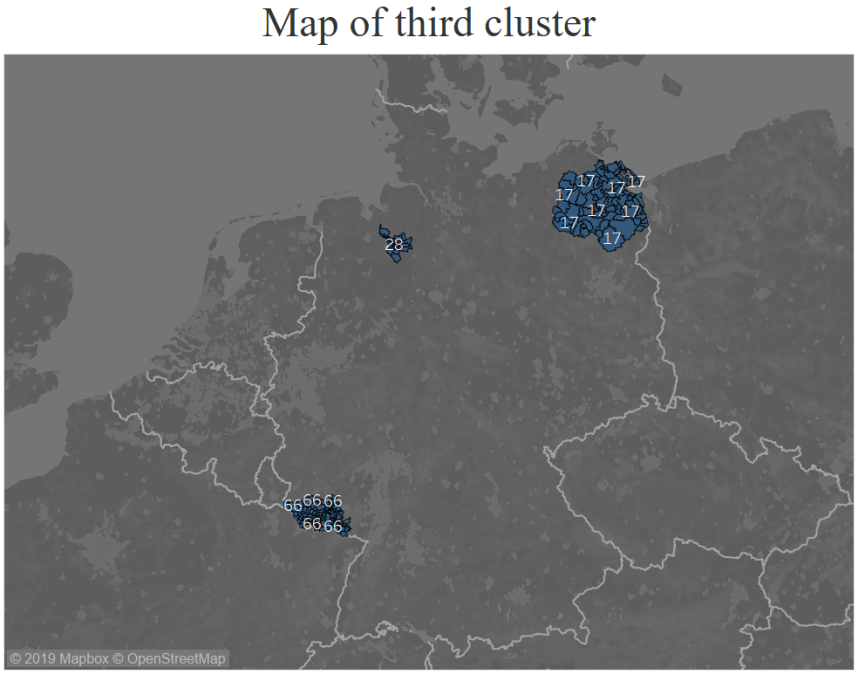


Figure 6.2.4: Visualization of the third cluster that is obtained when applying the agglomerative hierarchical clustering with average linkage to the mean of our time series data sets. It contains `pl` **17**, `pl` **28** and `pl` **66**.

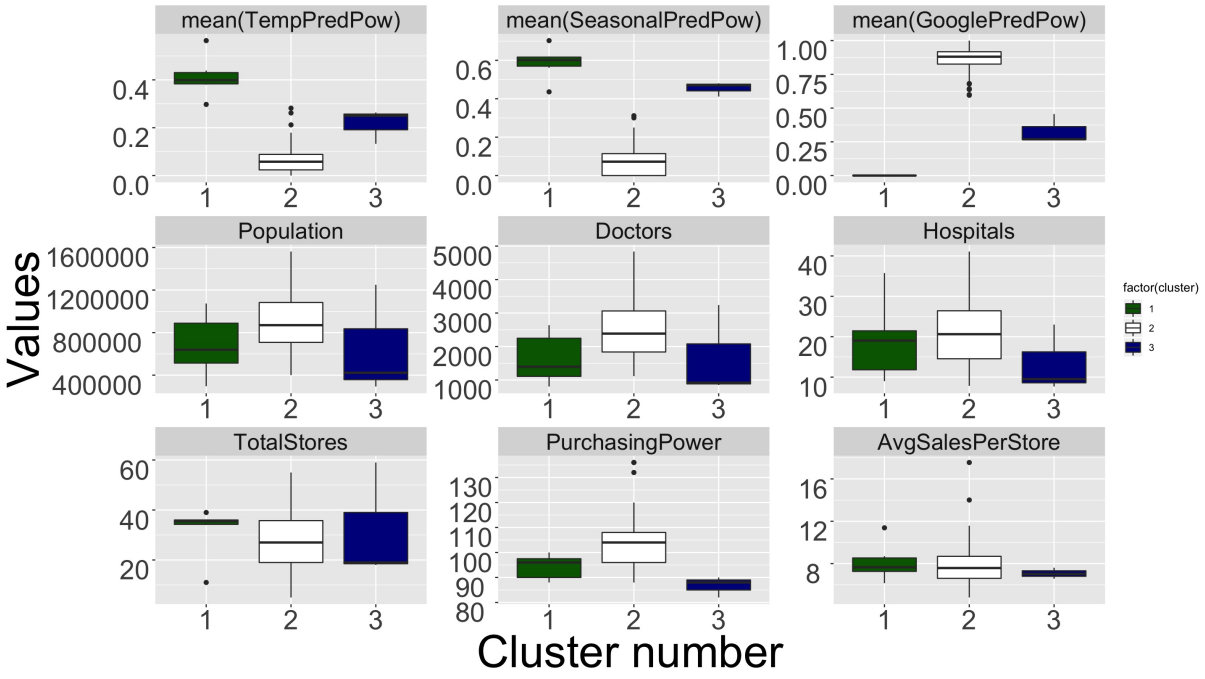


Figure 6.2.5: Boxplots of the clustering features and time independent variables per cluster that is obtained when applying the agglomerative hierarchical clustering with average linkage to the mean of our time series data sets.

Before we address the problem of dependence, we give an overview of the clusters in Table 6.2.4 .

Cluster	Total number of `pl` regions	Description
$cluster_1$	6	`pl` regions whose sales are explained by seasonal and temperature variables
$cluster_2$	86	`pl` regions whose sales are usually predicted by Google trends variable
$cluster_3$	3	`pl` regions whose sales are usually explained by all variables

Table 6.2.4: Overview of the three clusters that are obtained when applying the agglomerative hierarchical clustering with average linkage to the mean of our time series data sets.

Chapter 7

Dependence modelling

In the previous sections, we introduced our data set and explained how we fitted different models to predict the response, e.g. predict the weekly number of total sales in `pl` **77**. Next, we explored the model results and used them as an input for the clustering. Thus, we obtained a region segmentation for our data provider.

As the final step in the thesis, we work with copulas in this chapter so that we evaluate if there is a dependence between the pairs of `pl` regions. We remind that the theoretical background of our implementation can be found in Chapter 2 and for copulas one can look at Section 2.7. Moreover, the list of the `R` implementations that we apply in this chapter can be found below:

selm.fit function: Implementation available in the `sn` package given by Azzalini (2019).
pst function: Implementation available in the `sn` package.

7.1 u-data

One of the crucial steps to be able to use copulas in the analysis is to have the approximately uniformly distributed data set, called **u-data**, since it is the input for a copula model as we discussed in Section 2.7.

In our case because we have the standardized residuals $s_{tj_0p_0}(p_0 \in [1,10], j_0 \in [1,95],$ t = 1, ..., 169) from the final harmonic regression model, we can further work with these residuals, i.e. we can analyze if they are independent and how they are distributed. Thus, we start the analysis with checking if the residuals $s_{tj_0p_0}$ are satisfying the independency assumption. To do that, we look into the auto correlation values and visualize them via an autocorrelation function plot as can be seen in Figure 7.1.1. What we see in Figure 7.1.1, is that the autocorrelation value at lag 1 of `pl` **77** is above 0.3. Moreover, there are some other `pl` regions out of 95 regions, where the autocorrelation value at lag 1 seems to be above the threshold, shown by red line in Figure 7.1.1. To take the time effect out, we

therefore fit $AR(1)$ model to the residuals $s_{tj_0p_0}$ and obtain new residuals from the $AR(1)$ model, denoted by $r^{AR(1)}_{tj_0p_0}$. We define them as:

$$r^{AR(1)}_{tj_0p_0} := \frac{s_{tj_0p_0} - \hat{\rho} \cdot s_{(t-1)j_0p_0}}{\hat{\tau}_{s_{tj_0p_0}}} \quad \text{for} \quad t = 1, ..., 169, \quad j_0 \in [1, 95] \quad \text{and} \quad p_0 \in [1, 10], \tag{7.1.1}$$

where $\tau_{s_{tj_0p_0}}$ denotes the standard deviation of the residuals $s_{tj_0p_0}$ and ρ denotes the coefficient of the $AR(1)$ model.

As can be seen on the right plot of Figure 7.1.1, the residuals $r^{AR(1)}_{tj_0p_0}$ do not have the time dependency anymore.

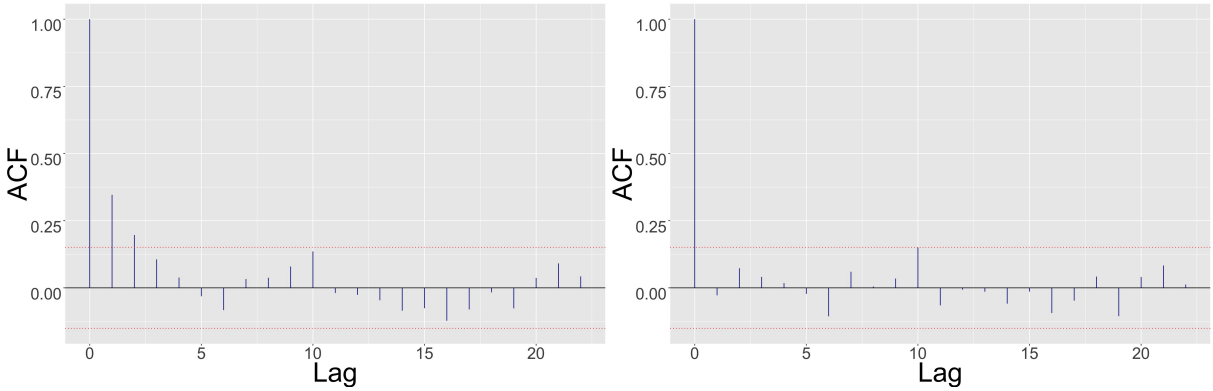


Figure 7.1.1: *Left*: Autocorrelation function plots of the residuals of harm.precise model for the `pl` **77**, *Right*: Autocorrelation function plots of the residuals of $AR(1)$ model fitted to the residuals of the harm.precise model for the `pl` **77**.

Moreover, we give the histogram of the estimated ρ values of the $AR(1)$ model in Figure 7.1.2. What we observe here, is that the estimated ρ values change between -0.04 and 0.50; furthermore, its' median is around 0.25. Because we did not account for the one-week time effect in our harmonic regression setup, these values seem to be reasonable.

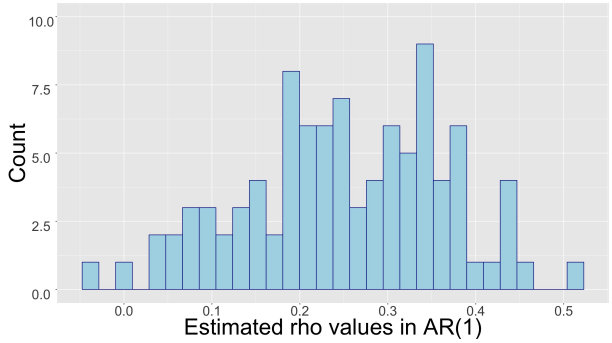


Figure 7.1.2: Histogram of the ρ values of the $AR(1)$ model, which is fitted to the standardized residuals of the harm.precise model.

As a next step, we investigate the distribution of the residuals $r^{AR(1)}_{tj_0p_0}$. Our ultimate goal here is to obtain u-data by using the probability integral transform as we discussed in Section 2.7.1. After analyzing density histograms, we conclude that the residuals are following the univariate skew-t distribution, introduced in Section 2.1. To show our conclusion, we add several plots. We first present the histogram of the simulated data by the univariate skew-t distribution with different parameters in Figure 7.1.4. When we compare it with Figure 7.1.3, we observe that the residuals $(r^{AR(1)}_{t38p_0}, r^{AR(1)}_{t75p_0})$ are approximately distributed by the univariate skew-t distributions. However, the distribution parameters of each `pl` region seems to be different.

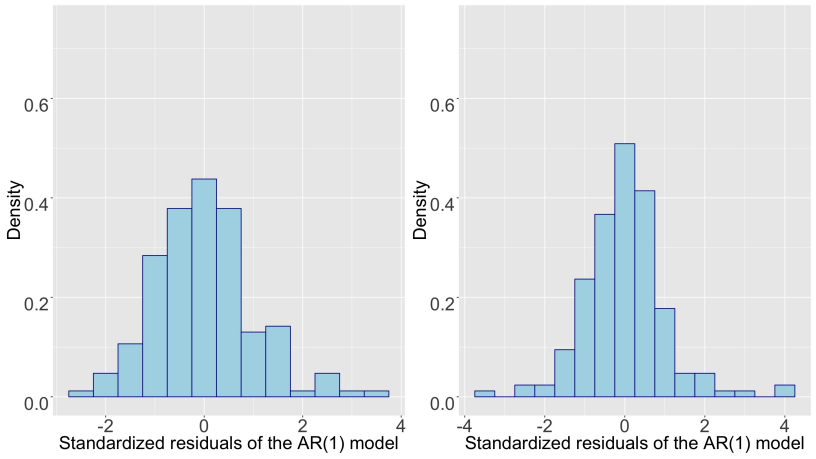


Figure 7.1.3: *Left*: Histogram of the residuals of `pl` **38**, $r^{AR(1)}_{t38p_0}$,*Right*: Histogram of the residuals of the `pl` **75,** $r^{AR(1)}_{t75p_0}$. They seem to follow the univariate skew-t distribution.

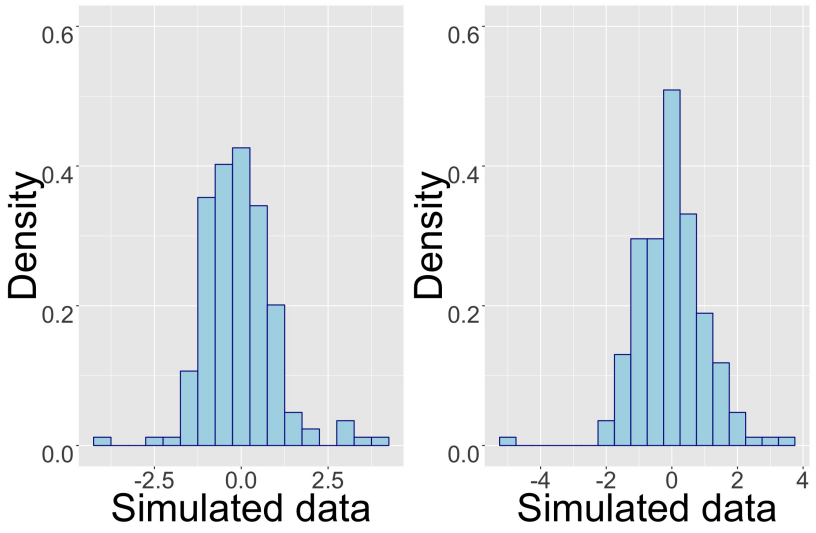


Figure 7.1.4: *Left*: Histogram of the data, simulated by the univariate skew-t distribution with the location parameter -0.75, the scale parameter 1.10, the skewness parameter 1.20 and the degree of freedom 12. *Right*: Histogram of the data, simulated by the univariate skew-t distribution with the location parameter -0.20, the scale parameter 0.70, the skewness parameter 0.30 and the degree of freedom 4.

Because we concluded that the residuals $r^{AR(1)}_{tj_0p_0}(p_0 = 1, j_0 \in [1, 95], t = 1, ..., 169)$ are following the univariate skew-t distribution with different parameters, for each `pl` region we need to optimize the distribution parameters, the location, scale and skewness parameters, as well as the degree of freedom. Hence, we use the *selm.fit* function from the `sn` package, which returns the optimized parameters by the maximum likelihood estimation. For instance, it returns the location parameter -0.21, the scale parameter 0.72, the skewness parameter 0.29 and the degree of freedom 3.83 for `pl` **75**, whereas the location parameter -0.75, the scale parameter 1.14, the skewness parameter 1.20 and the degree of freedom 12.21 for `pl` **38**.

As a final step for obtaining the u-data, we apply the probability integral transform to the residuals $r^{AR(1)}_{tj_0p_0}$, where the distribution is univariate skew t and the parameters of it are optimized for each `pl` region. What we obtain as the u-data, can be seen in Figure 7.1.5 for `pl` **75** and `pl` **38**. We observe that the estimated u-data is approximately uniformly distributed. To convince ourselves more, we present the comparison of our u-data with the simulated u-data in Figure 7.1.6. We see that the pdf of our u-data is highly similar to the pdf of the univariate standard uniform distribution. Hence, our u-data is reasonable to work further with copulas.

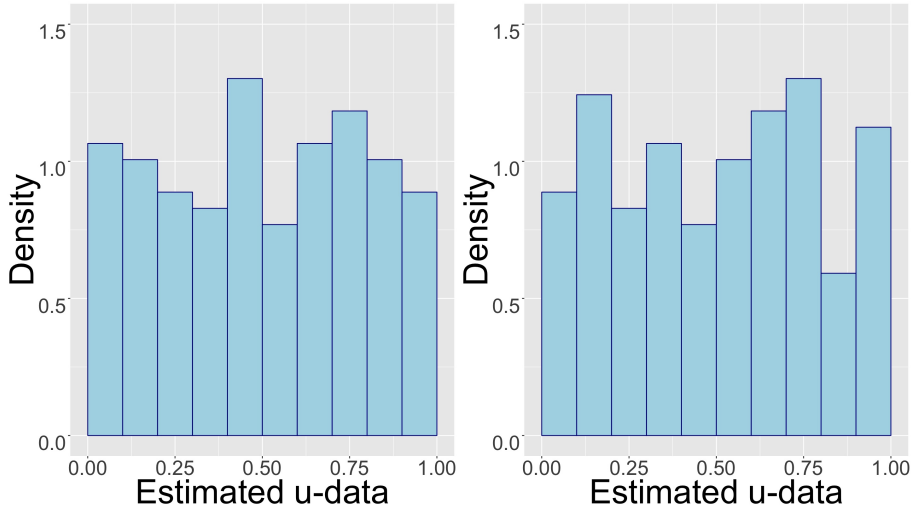


Figure 7.1.5: *Left*: Histogram of the estimated u-data of `pl` **75**, *Right*: Histogram of the estimated u-data of `pl` **38**.

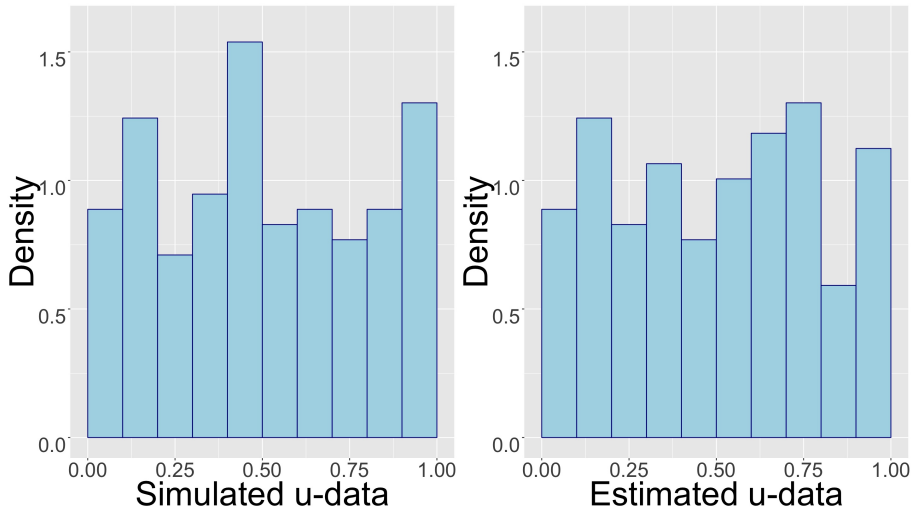
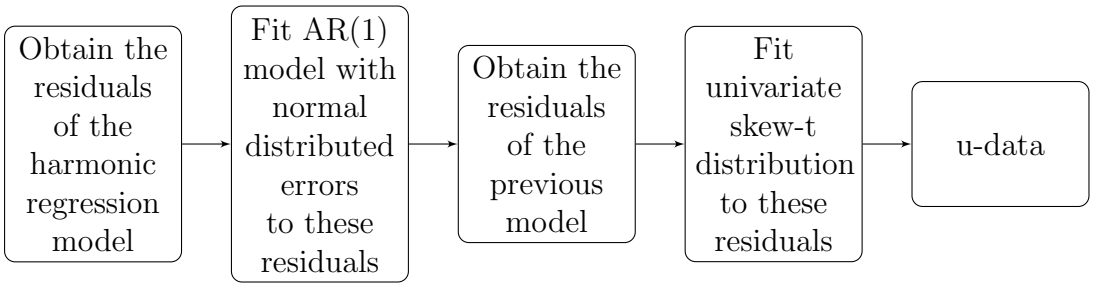


Figure 7.1.6: *Left*: Histogram of the 169 data points from the univariate standard uniform distribution, *Right*: Histogram of the estimated u-data of `pl` **38**. Our u-data is reasonable to use them inputs for a copula model.

To sum up, we reach the u-data as shown in the following flow chart.



7.2 Pairs plots

After obtaining u-data, we can look at pairs plots of each margin; however, we have 95 margins (`pl` regions) in total and we are mostly interested in the dependence between the close regions. Therefore, we continue our dependence analysis on the state level. We remind that Germany consists of 16 states; however, some states include only a few `pl` regions such as Saarland. On the other hand, there are more than 10 `pl` regions within several states like Bavaria. Hence, we analyze the states, which contains the small number of `pl` regions, together with their close states. Moreover, we separately look into the north and south parts of the states, whose total number of `pl` regions are high. As a result, the states that we together analyze as pairs are Brandenburg and Berlin, Lower Saxony and Bremen, Rhineland-Palatinate and Saarland; as triples are Schleswig-Holstein, Hamburg and Mecklenburg-Vorpommen; as well as, Saxony, Saxony-Anhalt and Thuringia. Furthermore, we separately focus on the north and south parts of Bavaria and North Rhine-Westphalia. The two states that we solitarily examine are Hessen and Baden-Württemberg. Before investigating the pairs plots, we give the summary of our region division in Table 7.2.1.

Area	State names	`pl` regions	Number of `pl` regions
1	Brandenburg, Berlin	`pl` 3, 10, 12, 13, 14, 15, 16	7
2	Baden-Württemberg	`pl` 69, 70, 71, 72, 73, 74, 75, 76, 77, 78, 79, 88	12
3	Bavaria	`pl` 80, 81, 82, 83, 84, 85, 86, 87, 89	9
4	Bavaria	`pl` 90, 91, 92, 93, 94, 95, 96, 97	8
5	Hessen	`pl` 34, 35, 36, 60, 61, 63, 64, 65, 68	9
6	Lower Saxony, Bremen	`pl` 21, 26, 27, 29, 30, 31, 37, 38, 49, 28	10
7	North Rhine-Westphalia	`pl` 32, 33, 44, 45, 46, 47, 48, 59	8
8	North Rhine-Westphalia	`pl` 40, 41, 42, 50, 51, 52, 57, 58	8
9	Rhineland-Palatinate, Saarland	`pl` 53, 54, 55, 56, 67, 66, 31, 37, 38, 239	6
10	Schleswig-Holstein, Hamburg, Mecklenburg-Vorpommen	`pl` 20, 22, 17, 18, 19, 23, 24, 25	8
11	Saxony, Saxony-Anhalt, Thuringia	`pl` 1, 2, 4, 6, 8, 9, 39, 99, 7, 98	10

Table 7.2.1: Overview of the areas that we analyze the dependence within them.

We now want to fit bivariate copula families to each pair of the `pl` regions of the given area. Here we make use of the *BiCopSelect* function from the `R` package `VineCopula` implemented by Schepsmeier et al. (2018). Note that we always use the AIC selection criterion. The bivariate copula families that we utilize in our analysis are Gaussian, Student's t, Clayton, Gumbel, Frank and Joe, as well as the 90,180 and 270 degrees rotations of these. We recall that the examples of the bivariate copula families and more details were given in Section 2.7. Next, we investigate how many times Gaussian or non-Gaussian copula is fitted and analyze the maximum and minimum values of the Kendall's τ. As a result, we will be able to discuss if the dependencies within each area is more likely Gaussian and how strong the dependencies are.

We present the fitting results in Table 7.2.2 and show the information about the maximum and minimum values of the Kendall's τ within each area in Table 7.2.3. We give the same information together in Table 7.2.4 for convenience. What we see, is that the there is usually non-Gaussian dependence between the `pl` regions. The majority of fitted families is given by symmetric tail dependent Student's t copulas. These are supplemented

by smaller numbers of asymmetric tail dependent Gumbel, Clayton and Joe copulas. However, the strength of the dependence is low, since we have low Kendall's τ values. Furthermore, the dominant bivariate copula family in the fourth and eleventh areas is Student's t, where the Kendall's τ values can take values up to 0.37 and 0.41, respectively. Additionally the third, fourth and eleventh areas have more tail dependence than the other regions, since the number of times that either Frank or Gaussian copula is chosen is too low compared to all number of fits. We recall that Gaussian and Frank copulas do not model any tail dependence.

Area	Total number of pairs	Number (percentage) of Gaussian fit	Number (percentage) of Student's t fit	Number (percentage) of Clayton fit	Number (percentage) of Frank fit	Number (percentage) of Gumbel fit	Number (percentage) of Joe fit
1	21	4 (19%)	7 (33%)	0 (0%)	9 (43%)	1 (5%)	0 (0%)
2	66	1 (2%)	16 (24%)	16 (24%)	9 (14%)	14 (21%)	10 (15%)
3	36	1 (3%)	7 (19%)	6 (17%)	3 (8%)	9 (25%)	10 (28%)
4	28	0 (0%)	14 (50%)	1 (4%)	4 (14%)	6 (21%)	3 (11%)
5	36	2 (5%)	9 (25%)	4 (11%)	11 (31%)	5 (14%)	5 (14%)
6	45	3 (7%)	11 (24%)	9 (20%)	6 (13%)	11 (24%)	5 (11%)
7	28	0 (0%)	9 (32%)	4 (14%)	9 (32%)	4 (14%)	2 (7%)
8	28	0 (0%)	12 (43%)	4 (14%)	9 (32%)	2 (7%)	1 (4%)
9	15	2 (13%)	8 (53%)	2 (13%)	2 (13%)	1 (7%)	0 (0%)
10	28	1 (4%)	7 (25%)	4 (14%)	11(39%)	2 (7%)	3 (11%)
11	45	0 (0%)	24 (53%)	5 (11%)	6 (13%)	7 (16%)	3 (7%)

Table 7.2.2: Number and percentage of occurrences of the bivariate copula families in each bivariate copula fit of the given area.

Area	$\min(Kendall's\,\tau)$	$\max(Kendall's\,\tau)$
1	0.14	0.37
2	0.12	0.40
3	0.15	0.36
4	0.19	0.36
5	0.11	0.32
6	0.13	0.37
7	0.07	0.36
8	0.19	0.36
9	0.14	0.34
10	0.09	0.41
11	0.11	0.41

Table 7.2.3: Minimum and maximum values of the Kendall's τ within each area.

Area	Total number of pairs	Number (percentage) of Gaussian fit	Number (percentage) of non-Gaussian fit			Kendall's τ	
			Symmetric tail dependence	Asymmetric tail dependence	No tail dependence	min	max
1	21	4 (19%)	7 (33%)	0 (5%)	9 (43%)	0.14	0.37
2	66	1 (2%)	16 (24%)	40 (60%)	9 (14%)	0.12	0.40
3	36	1 (3%)	7 (19%)	25 (70%)	3 (8%)	0.15	0.36
4	28	0 (0%)	14 (50%)	10 (36%)	4 (14%)	0.19	0.36
5	36	2 (5%)	9 (25%)	14 (39%)	11 (31%)	0.11	0.32
6	45	3 (7%)	11 (24%)	25 (55%)	6 (13%)	0.13	0.37
7	28	0 (0%)	9 (32%)	10 (35%)	9 (32%)	0.07	0.36
8	28	0 (0%)	12 (43%)	7 (25%)	9 (32%)	0.19	0.36
9	15	2 (13%)	8 (53%)	3 (20%)	2 (13%)	0.14	0.34
10	28	1 (4%)	7 (25%)	9 (32%)	11(39%)	0.09	0.41
11	45	0 (0%)	24 (53%)	15 (34%)	6 (13%)	0.11	0.41

Table 7.2.4: Number and percentage of occurrences of the Gaussian or non-Gaussian copula families, which are grouped by their tail dependence properties, in each bivariate copula fit of the given area and the minimum and maximum values of the Kendall's τ within each area.

In the following sections, both the regions within a Germany map and their pairs plots can be seen. As we stated above and can be seen in Figure 7.2.8 and Figure 7.2.22, we observe many symmetric tail dependencies especially within North Bavaria and Saxony, Saxony-Anhalt and Thuringia, which are the fourth and the eleventh areas in our analysis, respectively. In North Bavaria the symmetric tail dependence between the `pl` **90** and the `pl` **93** are strong, moreover the corresponding Kendall's τ value, 0.36, is the highest one among the Kendall's τ values of this area. On the other hand, the strongest dependence within Saxony, Saxony-Anhalt and Thuringia exists between the `pl` **1** and the `pl` **7**, where the the corresponding Kendall's τ value is 0.41; however, the dependence is asymmetric. Contrary, we usually do not see any tail dependence in Brandenburg and Berlin, which is the first area in our analysis. Figure 7.2.2 shows that the strongest dependence in this area exists between the `pl` **14** and `pl` **16** with the corresponding the Kendall's τ value, 0.37; but it does not contain any tail dependence.

Area 1: Brandenburg and Berlin

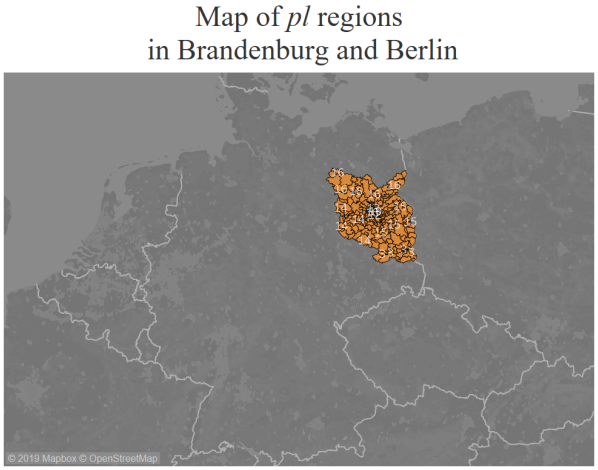


Figure 7.2.1: Visualization of the `pl` regions in Brandenburg and Berlin.

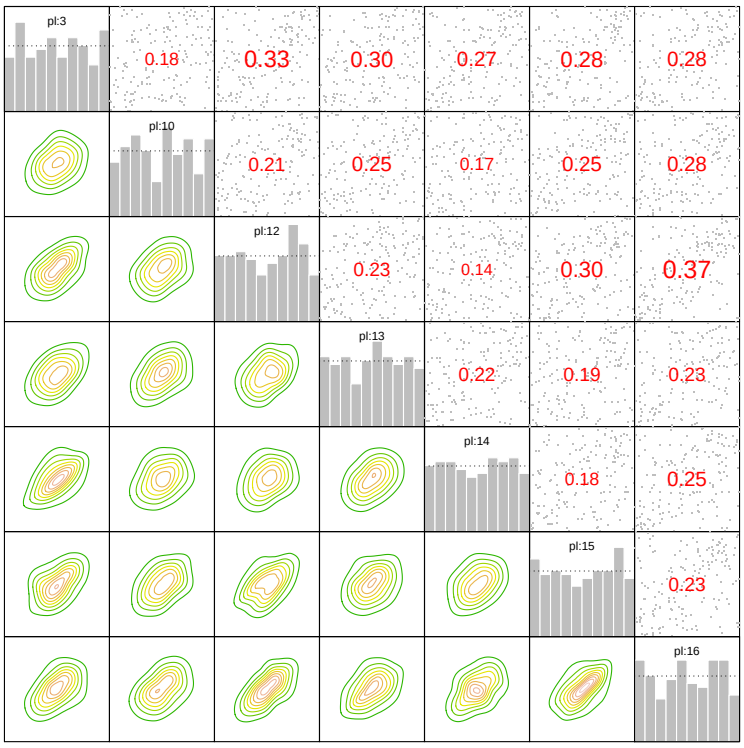


Figure 7.2.2: Brandenburg and Berlin: *Upper*: pairs plots of copula data, *diagonal*: histogram of copula margins, *lower*: normalized contour plots for the `pl` regions of the first area given in Table 7.2.1.

Area 2: Baden-Württemberg

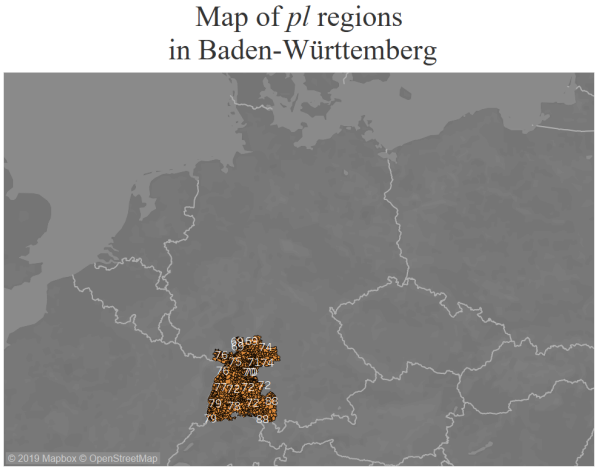


Figure 7.2.3: Visualization of the `pl` regions in Baden-Württemberg.

Figure 7.2.4: Baden-Württemberg: *Upper*: pairs plots of copula data, *diagonal*: histogram of copula margins, *lower*: normalized contour plots for the `pl` regions of the second area given in Table 7.2.1.

Area 3: South Bavaria

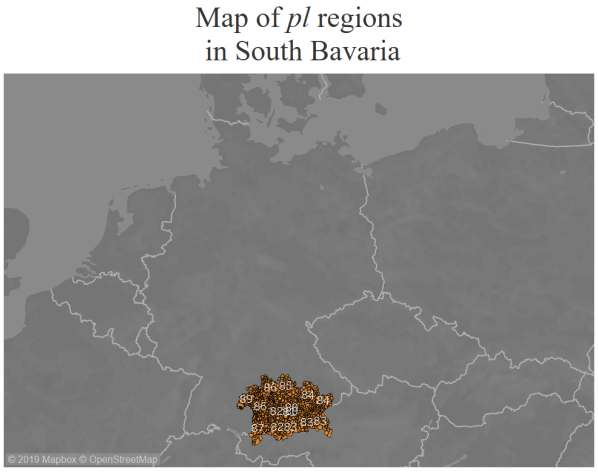


Figure 7.2.5: Visualization of the `pl` regions in South Bavaria.

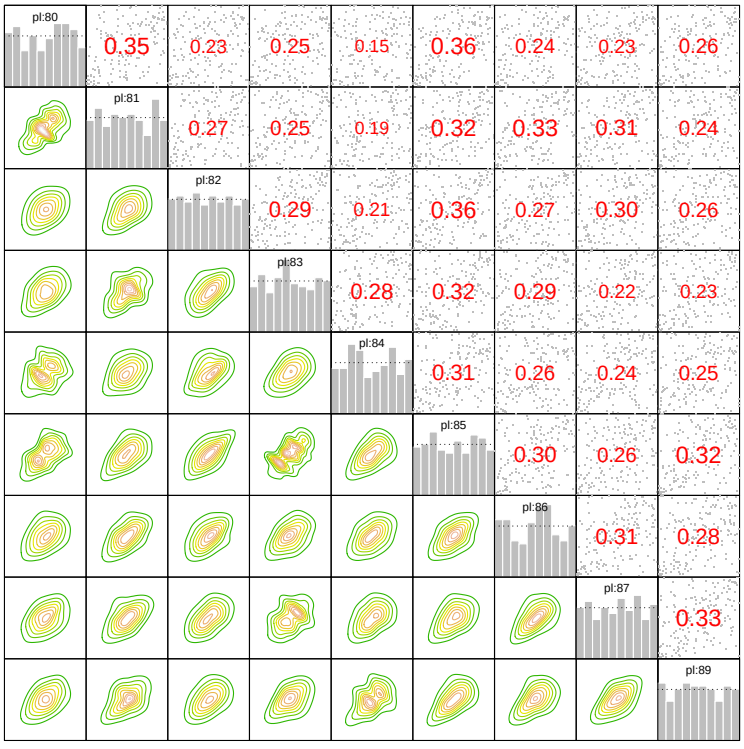


Figure 7.2.6: South Bavaria: *Upper*: pairs plots of copula data, *diagonal*: histogram of copula margins, *lower*: normalized contour plots for the `pl` regions of the third area given in Table 7.2.1.

Area 4: North Bavaria

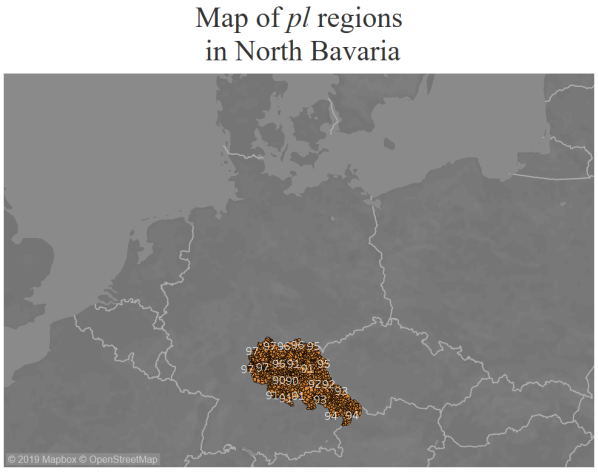


Figure 7.2.7: Visualization of the `pl` regions in North Bavaria.

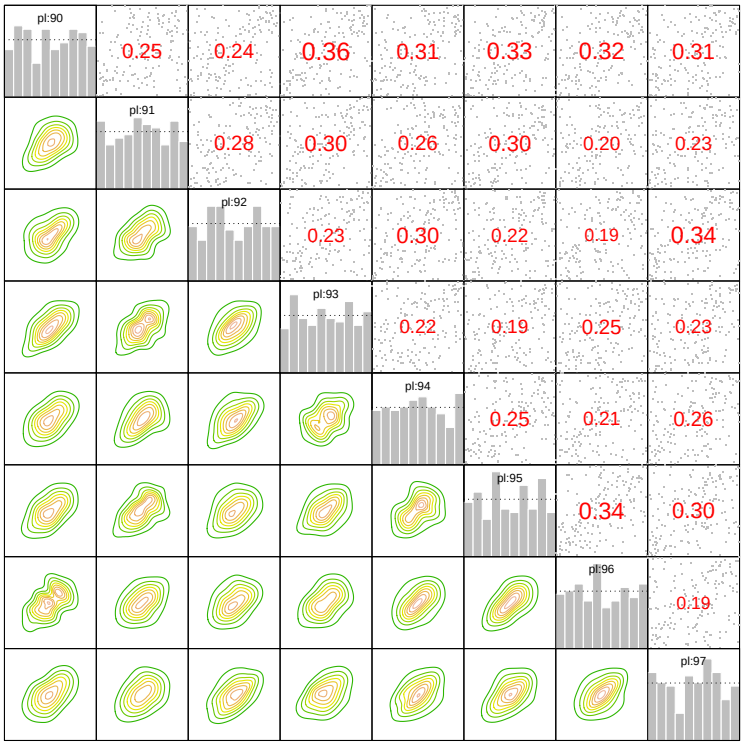


Figure 7.2.8: North Bavaria: *Upper*: pairs plots of copula data, *diagonal*: histogram of copula margins, *lower*: normalized contour plots for the `pl` regions of the fourth area given in Table 7.2.1.

Area 5: Hessen

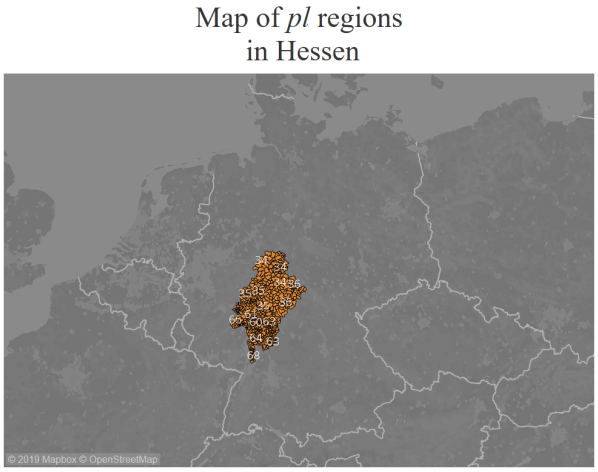


Figure 7.2.9: Visualization of the `pl` regions in Hessen.

Figure 7.2.10: Hessen: *Upper*: pairs plots of copula data, *diagonal*: histogram of copula margins, *lower*: normalized contour plots for the `pl` regions of the fifth area given in Table 7.2.1.

Area 6: Lower Saxony and Bremen

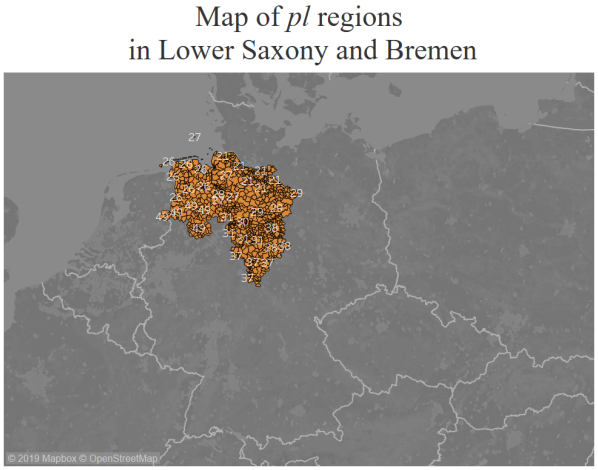


Figure 7.2.11: Visualization of the `pl` regions in Lower Saxony and Bremen.

Figure 7.2.12: Lower Saxony and Bremen: *Upper*: pairs plots of copula data, *diagonal*: histogram of copula margins, *lower*: normalized contour plots for the `pl` regions of the sixth area given in Table 7.2.1.

Area 7: North of North Rhine-Westphalia

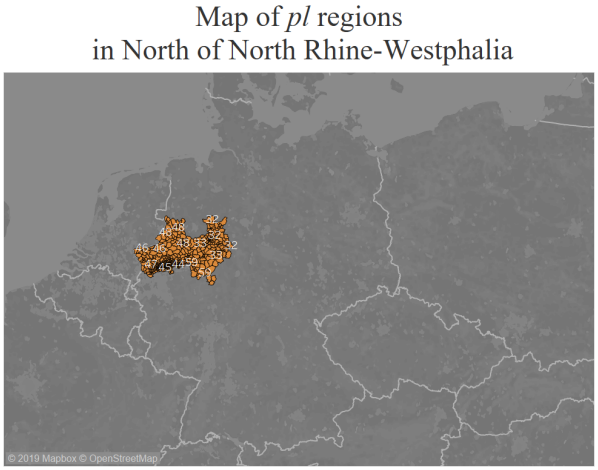


Figure 7.2.13: Visualization of the `pl` regions in North of North Rhine-Westphalia.

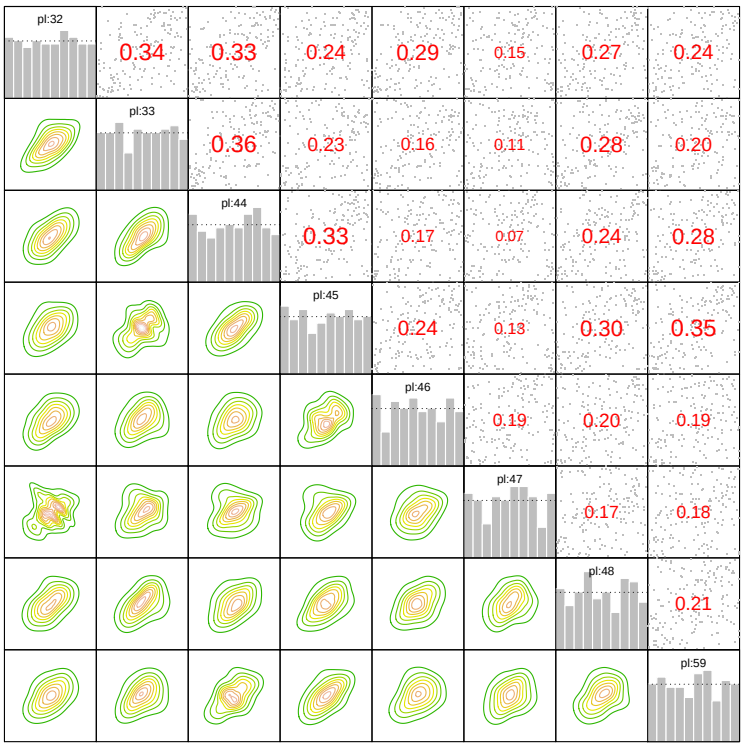


Figure 7.2.14: North of North Rhine-Westphalia: *Upper*: pairs plots of copula data, *diagonal*: histogram of copula margins, *lower*: normalized contour plots for the `pl` regions of the seventh area given in Table 7.2.1.

Area 8: South of North Rhine-Westphalia

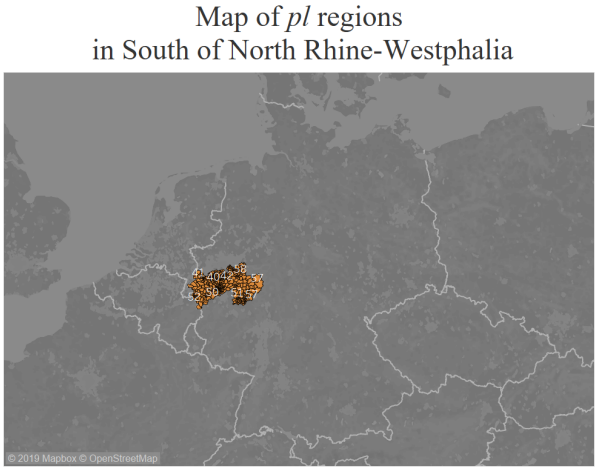


Figure 7.2.15: Visualization of the `pl` regions in South of North Rhine-Westphalia.

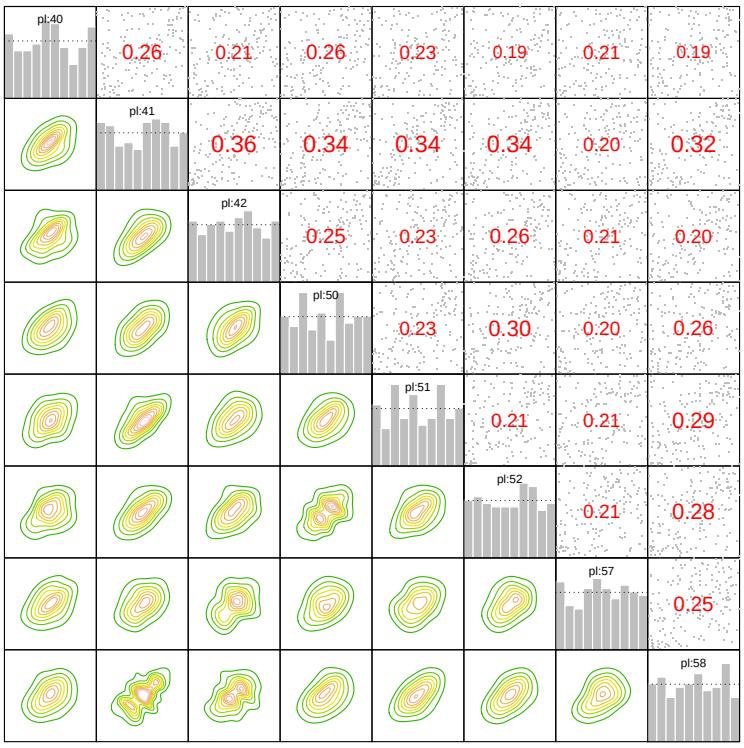


Figure 7.2.16: South of North Rhine-Westphalia: *Upper*: pairs plots of copula data, *diagonal*: histogram of copula margins, *lower*: normalized contour plots for the `pl` regions of the eighth area given in Table 7.2.1.

Area 9: Rhineland-Palatinate and Saarland

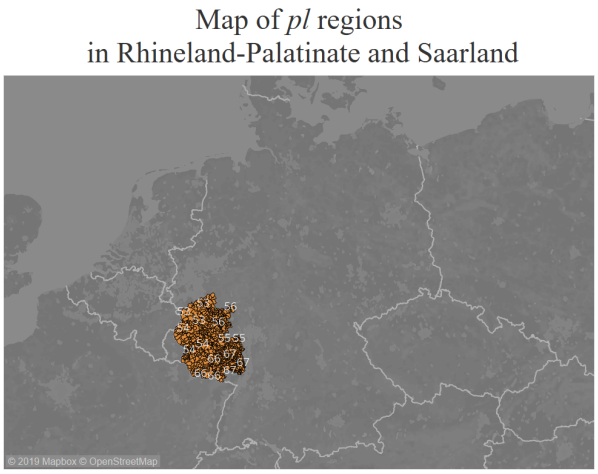


Figure 7.2.17: Visualization of the `pl` regions in Rhineland-Palatinate and Saarland.

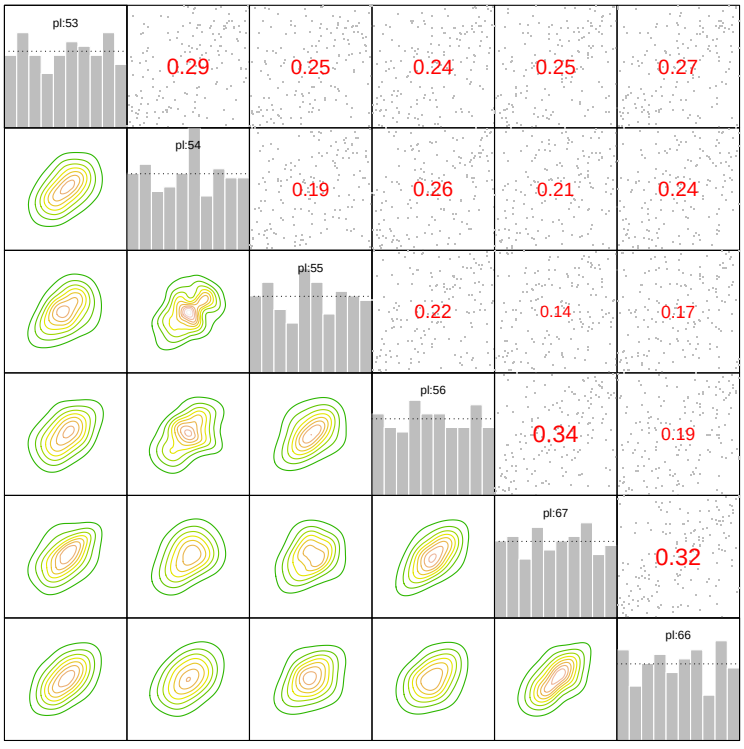


Figure 7.2.18: Rhineland-Palatinate and Saarland: *Upper*: pairs plots of copula data, *diagonal*: histogram of copula margins, *lower*: normalized contour plots for the `pl` regions of the ninth area given in Table 7.2.1.

Area 10: Schleswig-Holstein, Hamburg and Mecklenburg-Vorpommen

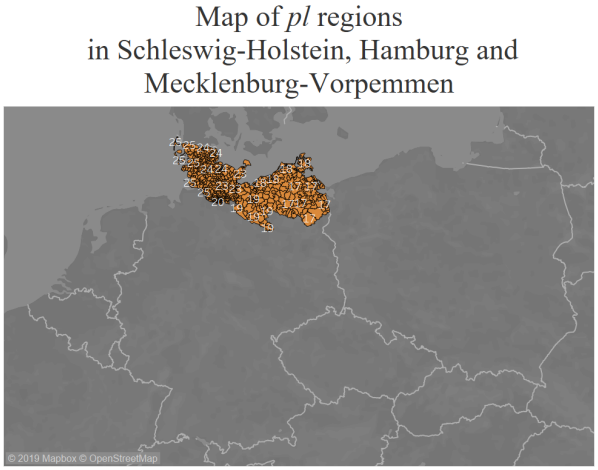


Figure 7.2.19: Visualization of the `pl` regions in Schleswig-Holstein, Hamburg and Mecklenburg-Vorpommen.

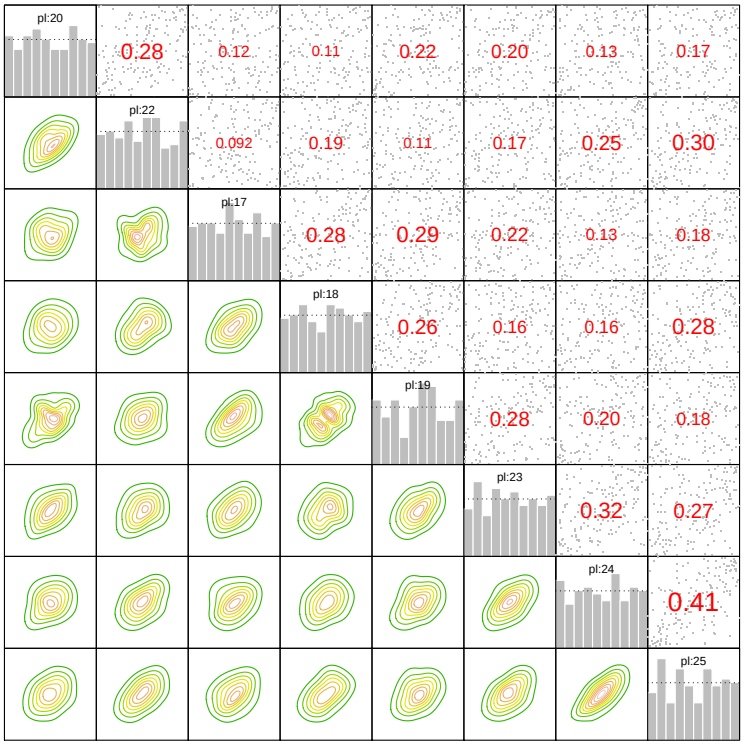


Figure 7.2.20: Schleswig-Holstein, Hamburg and Mecklenburg-Vorpommen: *Upper*: pairs plots of copula data, *diagonal*: histogram of copula margins, *lower*: normalized contour plots for the `pl` regions of the tenth area given in Table 7.2.1.

Area 11: Saxony, Saxony-Anhalt and Thuringia

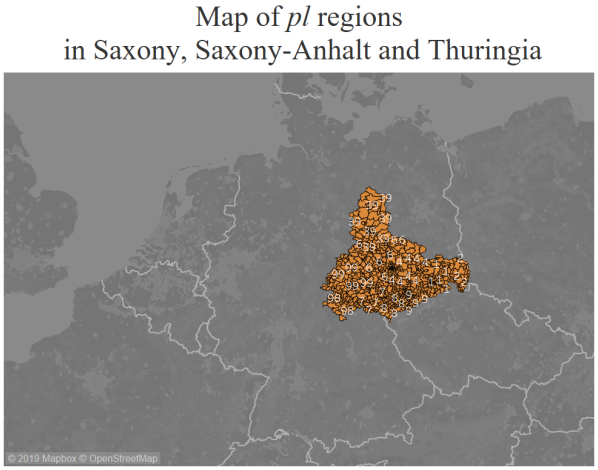


Figure 7.2.21: Visualization of the `pl` regions in Saxony, Saxony-Anhalt and Thuringia.

Figure 7.2.22: Saxony, Saxony-Anhalt and Thuringia: *Upper*: pairs plots of copula data, *diagonal*: histogram of copula margins, *lower*: normalized contour plots for the `pl` regions of the eleventh area given in Table 7.2.1.

Chapter 8

Conclusion

This thesis is concerned with the statistical analysis of the number of sales of a pharmacy product.

We discussed the necessary theory and applied various methods and models to our data. What we observed at the beginning, is that data preprocessing is a crucial step especially in the era of big data. In particular, the aggregation and outlier detection techniques can change the way of the analysis. Moreover, we showed that using the third party data from different sources can improve the forecasting performances. What we additionally argued, is that these third party data sets can have a meaning in the clustering analysis. In particular, we utilized the Google trends, the temperature data and the state holidays data in Germany together with the harmonic regression components for the marginal model fitting step. What we observed as a result of this step, is that the product sales could be explained by the Google trends data in most of the areas. We thus showed that the Google trends data can be a potential explanatory variable for the predictions of the pharmacy products. Using these model fitting results, we applied different distance-based clustering techniques. Our analysis found three clusters with different characteristics, which helped the company to create new strategies. For instance, the regions within one cluster do not include any Google trends data as an explanatory variable. As a result, the company prefers not having online advertisements in these regions. Even though we could draw conclusions from our clustering analysis, we lastly analyzed the dependence structure within the data with the copula approach. We showed that mostly non-Gaussian and asymmetric dependence between the regions exist. Hence, a clustering method that accounts for the dependence between the groups is needed.

The results that we obtained can be used to have an area segmentation by the company. The steps can be implemented similarly for different products and the findings can be compared.

Bibliography

Akaike, H. (1973). "Information theory as an extension of the maximum likelihood principle". In: *Second International Symposium on Information Theory, Akademiai Kiado*, pp. 267–281.

Arti, Michael, José G. Clavel, Mathias Hoffmann, and Dilip Nachane (2007). *Harmonic Regression Models: A Comparative Review with Applications*. Tech. rep. 333. Institute for Empirical Research in Economics: University of Zurich.

Azzalini, A. (2019). *The R package* ***sn****: The Skew-Normal and Related Distributions such as the Skew-t (version 1.5-4)*. Università di Padova, Italia. URL: http://azzalini.stat.unipd.it/SN.

Azzalini, A. and A. Capitanio (2003). "Distributions generated by perturbation of symmetry with emphasis on a multivariate skew t-distribution". In: *Journal of the Royal Statistical Society: Series B (Statistical Methodology)* 65.2, pp. 367–389. DOI: 10.1111/1467-9868.00391.

Baker, Frank B. and Lawrence J. Hubert (1975). "Measuring the Power of Hierarchical Cluster Analysis". In: *Journal of the American Statistical Association* 70.349, pp. 31–38.

Barnett, Vic and Toby Lewis (1994). *Outliers in Statistical Data*. John Wiley and Sons, Chichester.

Bloomfield, Peter (2000). *Fourier Analysis of Time Series. An Introduction*. John Wiley and Sons, Chichester. DOI: 10.1002/0471722235.

Brockwell, P. J. and R. A. Davis (2016). *Introduction to Time Series and Forecasting*. 3rd ed. Springer International Publishing. DOI: 10.1007/978-3-319-29854-2.

Charrad, Malika, Nadia Ghazzali, Véronique Boiteau, and Azam Niknafs (2014). "NbClust: An R Package for Determining the Relevant Number of Clusters in a Data Set". In: *Journal of Statistical Software* 61.6, pp. 1–36. URL: http://www.jstatsoft.org/v61/i06/.

Chatfield, C. and M. Yar (1988). "Holt-Winters Forecasting: Some Practical Issues". In: *Journal of the Royal Statistical Society* 37, pp. 129–140.

Czado, C. (2019). *Analyzing Dependent Data with Vine Copulas. A Practical Guide With R*. Vol. 222. Springer International Publishing. DOI: 10.1007/978-3-030-13785-4.

Durbin, J (1960). "The Fitting of Time Series Models". In: *Review of the International Statistical Institute* 28, pp. 233–243.

Dy, Jennifer G. and Carla E. Brodley (Dec. 2004). "Feature Selection for Unsupervised Learning". In: *J. Mach. Learn. Res.* 5, pp. 845–889. ISSN: 1532-4435. URL: http://dl.acm.org/citation.cfm?id=1005332.1016787.

Everitt, B., S. Landau, M. Leese, and D. Stahl (2011). *Cluster Analysis*. 5th ed. John Wiley and Sons. ISBN: 978-0-470-74991-3.

Fox, J. (1997). *Applied regression analysis, linear models, and related methods*. US: Sage Publications, Inc.

Friedman, J. (1984). *A variable span smoother*. Tech. rep. 5. Laboratory for Computational Statistics: Department of Statistics, Stanford University.

Han, Jiawei, Micheline Kamber, and Jian Pei (2011). *Data Mining: Concepts and Techniques*. 3rd. Morgan Kaufmann Publishers Inc., pp. 443–543. ISBN: 0123814790, 9780123814791.

Hartigan, J. A. and M. A. Wong (1979). "A k-means clustering algorithm". In: *JSTOR: Applied Statistics* 28.1, pp. 100–108.

Hastie, T., T. Tibshirani, and J. Friedman (2009). *The Elements of Statistical Learning. Data Mining, Inference, and Prediction, Second Edition*. Springer-Verlag New York. DOI: 10.1007/978-0-387-84858-7.

Hunter, J. D. (2007). "Matplotlib: A 2D graphics environment". In: *Computing in Science & Engineering* 9.3, pp. 90–95. DOI: 10.1109/MCSE.2007.55.

Hyndman, Rob and George Athanasopoulos (2013). *Forecasting: principles and practice*. OTexts. URL: https://www.otexts.org/fpp.

Hyndman, Rob et al. (2019). *forecast: Forecasting functions for time series and linear models*. R package version 8.7. URL: http://pkg.robjhyndman.com/forecast.

Joe, H. (1997). *Multivariate Models and Dependence Concepts*. London: Chapman and Hall.

Jones, Eric, Travis Oliphant, Pearu Peterson, et al. (2001). *SciPy: Open source scientific tools for Python*. URL: http://www.scipy.org/.

Kendall, M. G (1938). "A new measure of rank correlation". In: *Biometrika* 30.1/2, pp. 81–93.

Kevin, Dobbin and Richard Simon (Apr. 2011). "Optimally splitting cases for training and testing high dimensional classifiers". In: *BMC Med Genomics* 31. DOI: 10.1186/1755-8794-4-31.

Kleinberg, Jon (2002). "An Impossibility Theorem for Clustering". In: *Proceedings of the 15th International Conference on Neural Information Processing Systems*. NIPS'02. Cambridge, MA, USA: MIT Press, pp. 463–470. URL: http://dl.acm.org/citation.cfm?id=2968618.2968676.

Koopmans, Lambert H. (2004). *The Spectral Analysis of Time Series*. Elsevier Inc. DOI: 10.1016/B978-0-12-419251-5.X5000-5.

Kotz, S. and S. Nadarajah (1995). *Multivariate T-Distributions and Their Applications*. Cambridge: Cambridge University Press.

Loader, Clive (1999). *Local Regression and Likelihood*. Springer-Verlag New York. DOI: 10.1007/b98858.

Mahajan, Meena, Prajakta Nimbhorkar, and Kasturi Varadarajan (2009). "The Planar k-Means Problem is NP-Hard". In: *Proceedings of the 3rd International Workshop on Algorithms and Computation*. WALCOM '09. Berlin, Heidelberg: Springer-Verlag, pp. 274–285. ISBN: 978-3-642-00201-4. DOI: 10.1007/978-3-642-00202-1_24.

May, R.J., H.R. Maier, and G.C. Dandy (2010). "Data splitting for artificial neural networks using som-based stratified sampling". In: *Neural Networks* 23, pp. 283–294.

McKinney, Wes (2010). "Data Structures for Statistical Computing in Python". In: *Proceedings of the 9th Python in Science Conference*. Ed. by Stéfan van der Walt and Jarrod Millman, pp. 51–56.

Milligan, G.W. and M.C. Cooper (1985). "An examination of procedures for determining the number of clusters in a data set". In: *Psychometrika* 50.2, pp. 159–179.

Nanopoulos, Alex, Rob Alcock, and Yannis Manolopoulos (2001). "Feature-based Classification of Time-series Data". In: *International Journal of Computer Research* 10, pp. 49–61.

Nelsen, R. B. (2006). *An introduction to copulas*. 2nd ed. Springer-Verlag New York. DOI: 10.1007/0-387-28678-0.

Oliphant, Travis (2006). *NumPy: A guide to NumPy*. USA: Trelgol Publishing. URL: http://www.numpy.org/.

Pedregosa, F. et al. (2011). "Scikit-learn: Machine Learning in Python". In: *Journal of Machine Learning Research* 12, pp. 2825–2830.

R Core Team (2018). *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing. Vienna, Austria. URL: https://www.R-project.org/.

Reitermanova, Z. (2010). "Data splitting". In: *WDS'10 Proceedings of Contributed Papers*, pp. 31–36.

Rousseeuw, Peter (Nov. 1987). "Silhouettes: A Graphical Aid to the Interpretation and Validation of Cluster Analysis". In: *J. Comput. Appl. Math.* 20.1, pp. 53–65. ISSN: 0377-0427. DOI: 10.1016/0377-0427(87)90125-7.

Schepsmeier, Ulf, Jakob Stoeber, Eike Christian Brechmann, Benedikt Graeler, Thomas Nagler, and Tobias Erhardt (2018). *VineCopula: Statistical Inference of Vine Copulas*. R package version 2.1.8. URL: https://CRAN.R-project.org/package=VineCopula.

Seabold, Skipper and Josef Perktold (2010). "Statsmodels: Econometric and statistical modeling with python". In: *9th Python in Science Conference*.

Sklar, A. (1959). "Fonctions de Répartition à n Dimensions et Leurs Marges". In: *Publications de l'Institut Statistique de l'Université de Paris* 8, pp. 229–231.

Telgarsky, Matus and Andrea Vattani (2010). *Hartigan's Method: k-means Clustering without Voronoi*.

Theil, H. (1961). *Economic forecasts and policy*. Contributions to economic analysis. North-Holland Pub. Co.

Venables, W.N. and B.D. Ripley (2002). *Modern Applied Statistics with S*. 4th ed. Springer-Verlag New York, p. 402. DOI: 10.1007/978-0-387-21706-2.

Wang, Xiaozhe, Anthony Wirth, and Liang Wang (2007). "Structure-Based Statistical Features and Multivariate Time Series Clustering". In: *Proceedings of the 2007 Seventh IEEE International Conference on Data Mining*. ICDM '07. IEEE Computer Society, pp. 351–360. ISBN: 0-7695-3018-4. DOI: 10.1109/ICDM.2007.103.

Warren Liao, T. (Nov. 2005). "Clustering of Time Series Data-a Survey". In: *Pattern Recogn.* 38.11, pp. 1857–1874. ISSN: 0031-3203. DOI: 10.1016/j.patcog.2005.01.025. URL: http://dx.doi.org/10.1016/j.patcog.2005.01.025.

Wickham, Hadley (2016). *ggplot2: Elegant Graphics for Data Analysis*. Springer-Verlag New York. ISBN: 978-3-319-24277-4. URL: https://ggplot2.tidyverse.org.