D I P L O M A R B E I T

Development and Implementation of Statistical Indicators for the Assessment of Brain Tumors

Ausgeführt am

Institut für Statistik und Wahrscheinlichkeitstheorie

der Technischen Universität Wien

unter der Anleitung von

Ao.Univ.Prof. Dipl.-Ing. Dr.techn. Peter Filzmoser

und

Univ.Ass. Dipl.-Ing. Dr.techn. Matthias Templ

durch

Semagül AKLAN

Mariahilferstr. 192/3/1/5

1150 Wien

Wien, 02. Mai 2012

_____

Unterschrift

Abstract

Basing on a request of a team of scientists from the Neurology department of the General Hospital Vienna (AKH), the topic of statistical support on the scope of computer-based assessment of pathological characteristics of brain tumors was covered in this diploma thesis.

The goal of this work was to investigate, if there exist any statistical methods, which can be used as indicators for the computer-based analysis and assessment of cell activities in human brain tumors. This matter was discussed by defining of four indicators and applying those on two samples of digitalized human brain tumor tissue sections.

This thesis focuses on two issues. On the one side, any information about the possible distribution or at least about the properties of the measurements is tried to accomplish. On the other side the spatial distribution of potential groupings is of peculiar interest.

Starting with a two dimensional kernel density estimation with a Gaussian kernel on a square grid, the defined indicators are applied and the obtained results are demonstrated.

Acknowledgements

I want to dedicate this diploma thesis to my family, especially to my parents Gültaze and Mahir. They have enabled me to study, have always supported me morally and gave me the motivation I needed to finish this work. Thank you so much mom and dad!

Here I want to express my thanks to my professor Ao.Univ.Prof. Dipl.-Ing. Dr. techn. Peter Filzmoser for his support. My special thanks goes to my thesis adviser Univ.Ass. Dipl.-Ing. Dr.techn. Matthias Templ, who has been available for me with his advice at any time. Thank you for your helpful suggestions and your patience.

I also want to thank the Medical University of Vienna, specially MD Matthias Preusser (Department of Internal Medicine/Oncology) and MD Johannes A. Hainfellner (Institut of Neurology), for their supply with the necessary data and information.

Finally a special thank-you dues to my friends, who have always been there for me.

Contents

Chapter 1

Introduction

While the individual man is an insoluble puzzle, in the aggregate he becomes a mathematical certainty. You can, for example, never foretell what any one man will be up to, but you can say with precision what an average number will be up to. Individuals vary, but percentages remain constant. So says the statistician.

Arthur Conan Doyle - *The Sign of the Four*

1.1 Motivation

In Austria the second most common cause of death is cancer. Each year cancer is diagnosed for about 38.000 people in Austria. There exist different types of cancer. The most frequent cancer cases are bowel, lung, breast and prostate cancer and the chance to get one of these cancer types before the age of 75 is about 10%. [Austria, 2011b]

Brain tumors are one of the most common type of central nervous system cancer. In 2009 the rate of malignant brain tumors was 1.6% of all reported new incidences of cancer in Austria, while the mortality rate was about 2.6% of all death rates of cancer. The chance to get a brain tumor was about 0.5% for men and about 0.4% for women. [Austria, 2011a]

There are various kinds of research and considerable improvements in the field of treatment of brain tumors and the death rates show a decrease regarding the past twelve years. But there is also a need of more research whithin the scope of computer-based assessment of pathological characteristics of brain tumors.
Therefore a team of scientists from the Neurology department of the AKH Vienna asked for statistical support on this issue. With this diploma thesis the first steps in this direction

of research will have been done.

1.2 Goal of the Thesis

The classification of tumors plays an important role for the treatment and is necessary for a harmonized evaluation and a simplified exchange of results. Moreover it is an important factor for the further prognosis.
The histological classification of tumors distinguishes three types of tumor differentiation: light-microscopical, electron-microscopical and immunhistochemical differentiation. In this thesis the last one will be considered.
The immunhistochemical tumor differentiation deals with the growth behavior, the differentiation and the metastatic spread of tumors. [Bertolini, 2012]

The metastatic spread is a purpose of staging of tumors, which deals with the anatomic spread and will not be considered during this thesis.
For the histomorphological analysis the typing and grading of tumors play an important role. This will be explained in detail in Chapter 2.

The cell proliferation is used as an index for the growth behaviour of tumors. It is an important factor for the prediction of the survival time of a patient. As yet the determination of the proliferation index has been occured manually. Since the computation depends on the investigator, the evaluation of the proliferation index appears to be a subjective method.

The goal of this diploma thesis is now to investigate if there exist any statistical methods which can be used as indicators for the computer-based analysis and assessment of cell activities in human brain tumors. The aim is to define indicators which ensure an objective assessment and which are also precise. During this work two samples of human brain tumors will be investigated.

1.3 Approach to Achieve the Goals

At first the scanned and digitalized brain tumor samples undergo a process of segmentation in several parts and a process of determination of the marked cell nuclei by the software product developed by DI Andreas Walser during his master's thesis, see [Walser, 2011].

The result is an **ASCII**-file for each segment, which contains - among other things - the information about the x and y coordinates of the marked cell nuclei.

Within this work all parts of the samples are imported and a bivariate kernel density estimation with a Gaussian kernel is conducted for each sector. This is necessary since the amounts of pixels even of the several parts of the data are too big for further analysis.

This thesis focuses on two issues. On the one side, any information about the possible distribution or at least about the properties of the measurements is tried to accomplish. On the other side the spatial distribution of potential groupings is of peculiar interest.

For the statistical analysis of data the free and open-source software environment **R**, version **2.14.2** is used, which is an object-oriented and interpreted language and environment for statistical computing and graphics. For further information about the software see [R, 2011].

1.4 Overview of the Diploma Thesis

Here an overview on the following chapters of the thesis will be given:

* In Chapter 2 the necessary medical background is presented.
* Chapter 3 gives an overview about some computer-based image analysis methods based on studies and in addition the editing of data for this thesis is described.
* In Chapter 4 the essential mathematical background, which is used for the determination of the indicators, is explained in more detail.
* In Chapter 5 the indicators are defined and explained.
* Chapter 6 shows the obtained results of the considered samples with a corresponding interpretation of the results.
* Chapter 7 will give a summary and a conclusion of the thesis and the results.

Chapter 2

Medical Background

2.1 Introduction to Brain Tumors

A tumor is a neoplasm, which is a lesion, a mass of cells, that is either be formed by an abnormal growth of neoplastic cells or is present at birth. In other words, they occur as a result of mutation and damage of the regulation of cell growth and allow cells to grow and proliferate out of control. The growth of a tumor can be affected by variant growth factors, the mitosis rate, the loss of cells and by the blood supply of the tumors.
Tumors can appear anywhere in the body, whereby a tumor is not necessarily equal to cancer. A cancer is a malignant tumor, whereas tumors can also be benign. [NINDS, 2011, Bertolini, 2012]

Benign tumors are not cancerous and consist of cells that are similar to normal cells. They have a slow growth and there is no spread into other parts of the body. Usually they can be removed surgically and mostly they do not reappear. [NINDS, 2011]

Malignant tumors are the cancerous ones which consist of cells that are different from normal cells. This kind of tumors are difficult to remove completely with surgery because of their unclear shape and their invasion to other surrounding tissue. [NINDS, 2011]

One of the most common types of cancer of the central nervous system are brain tumors. A brain tumor can occur within the brain or the central spinal canal and can be life-threatening because it can show an invasive and infiltrative character. There are var-

ious types of brain tumors. The ones within the brain itself, also called as intracranial tumors, arise commonly from neurons or glial (non-neuronal) cells, such as astrocytes, oligodentrocytes and ependymal cells. The glial cells are the cells which are responsible for regulating the internal environment and maintaining a stable constant condition of properties of neural cells. They also have a support and protection function for neurons in the brain as well as in other parts of the nervous system. [UK, 2012], [ASCO, 2012]

2.2 Classification of Tumors

There are several types of classification of central nervous system tumors. Tumors are differentiated by their localisation, their typing, their grading and their anatomic spread. The anatomic spread distinguishes if it is a primary tumor or a local lymph node or a distant metastasis.
The histomorphology deals with the typing and grading of tumors. The histological typing of tumors is structured through the similarity to the normal tissue. [Bertolini, 2012]

Central nervous system tumors are histopathologically graded commonly by the grading system of the World Health Organization established in 1993. This grading is based on the location and the cell-building of tumor cells. [NINDS, 2011]:

- **Grade I:** Tumors which have a slow growth and do not show a metastic spread belong to this group. These tumors are benign and the surgical removal of the entire tumor is often possible. Grade I tumors are associated with long-term survival.
- **Grade II:** Tumors of this type show a slow growth too, but they can infiltrate surrounding tissue and thus recur as higher grade tumors. These tumors can show a benign or malignant condition and the treatment depends on the location of the tumor. It can require chemotherapy, radiation and also surgery.
- **Grade III:** These tumors are malignant and often recur as higher grade tumors. The invasion and infiltration into other tissues is very quick. The treatment requires often a combination of chemotherapy, radiation and/or surgery.
- **Grade IV:** Tumors of this type are malignant and very aggressive. They show a very different makeup of tissue than the surrounding ones and invade rapidly other tissues. It is very hard to treat these tumors and an aggressive treatment is required.

2.3 Ependymomas

Ependymomas are primary glial brain tumors that arise from ependymal cells, which are tissues of the central nervous system. Between 3% and 9% of all neuroepithelial tumors are ependymomas and about 50% to 70% of ependymomas are located within the brain itself.
Children and young adults are mostly affected by this type of cancer, whereby in children about 90% of ependymomas are intracranial and usually occur in the infratentorial compartement, which is also known as the fourth ventricle of the brain.
By contrast, about 75% of ependymomas in adults and adolescents are located within the spinal canal and only a small number is intracranially and occurs in the supratentorial compartement of the brain.
The WHO-classification for these tumors distinguishes four types of ependymomas. The myxopapillary ependymoma (MPE) and the subependymoma are from WHO Grade I and about 85% of ependymomas are benign MPEs. Subependymomas are uncommon lesions, but show the benign characteristics of MPEs and affect usually adults over 40 years of age. From WHO Grade II are ependymomas with cellular, papillary and clear cell variants. Ependymomas can also be anaplastic and these ones are classified from type WHO Grade III. Malignant ependymomas and ependymoblastomas belong to this group.
The treatment of patients with ependymomas is influenced by the tumor grades and consists of neurosurgical intervention and post-operative radio- and/or chemotherapy. The low-grade ependymomas are usually treated with radiation therapy only, but in general a total surgical removal is preferred, in combination with radiation and chemotherapy. [Bruce, 2009], [Preusser et al., 2008].

The following images show an example from different views for an infratentorial ependymoma in a pediatric case. In the first image, Figure 2.1, the ependymoma is illustrated by an MRI (Magnetic Resonance Imaging) image through a sagittal view, which means that the image has been taken while a vertical plane passed through the standing body from the front to the back. [CERN, 2011b]

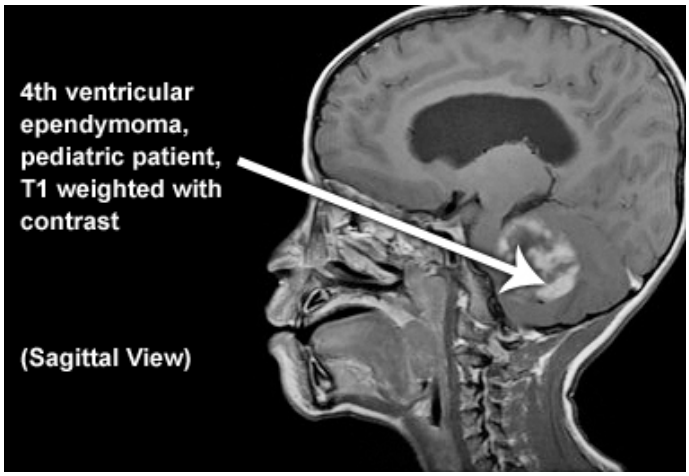

Figure 2.1: MRI-Sagittal View: Infratentorial Ependymoma, Source: [CERN, 2011a]

The second MRI-image, Figure 2.2, has been taken from an axial view, through the passing of a straight line through a spherical body between two poles and the body revolving around. [CERN, 2011b]

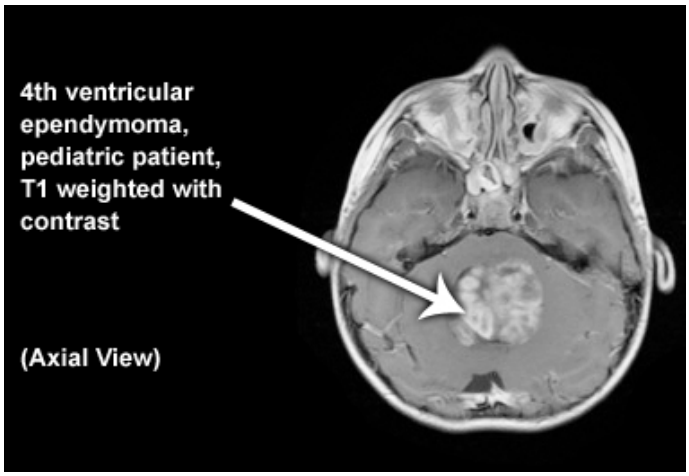

Figure 2.2: Axial View: Infratentorial Ependymoma, Source: [CERN, 2011a]

2.4 Materials and Methods

The materials used in this thesis are two of 78 specimens of intracranial ependymomas, which had been collected from a group of scientists from the General Hospital Vienna (AKH) primarily for diagnostic purposes, but which then also had been used for research purposes including the assessment of the Ki67 index. Ki67 is a histopathological biomarker and is used to determine the tumor cell proliferation.

These specimens had been taken from patients - ranged from 1.2 months to 74.4 years of age - at the AKH between 1965 and 1999 and had been selected after an initially evaluation of the quality and size of the tissue sections by the scientists J.A. Hainfellner and M. Preusser, where the ones with a small tissue size were excluded. The remaining 78 tissues had a size greater than one microscopic field at a magnification of x400. [Preusser et al., 2008]

2.4.1 Immunhistochemistry

After fixing the tumor tissue blocks with formalin and embedding those with paraffin, sections were cut at a thickness of 3-5 μm. Then a heat-induced epitope retrieval in 0.01 M (molar mass) citrate buffer (pH 6.0) was conducted with the slides for 30 minutes in a microwave oven at 600 W. After the incubation of the sections with a monoclonal mouse anti-Ki67 antibody at a dilution of 1:50 for 25 minutes, the detection process of the immunoreactivity using the ChemMate kit (Dako) and diaminobenzidine as chromogen was performed. The Ki67 immunohistochemistry was conducted according to the standard operating procedure of the laboratory of the AKH. [Preusser et al., 2008]

2.4.2 Computation of the Cell Proliferation

The conventional determination of the tumor cell proliferation index is in the following way:
The anti-Ki67 immunoreactive tissue section is scanned at a low magnification and the area with the highest density of immunolabelled tumor cell nuclei is determined. This area is also called as “hot-spot”. Then a total of 500 tumor cell nuclei are evaluated within the hot-spot area and through manual counting on an eye-grid the fraction of the labelled cell nuclei per 500 tumor cell nuclei is calculated and is expressed as a percentage. The count-

ing of 500 cells per case yields good results since it takes two minutes by an experienced person. Using a higher number of cells has been to tiresome in a routine setting. [Preusser et al., 2008]

Chapter 3

Computer-Based Digital Image Analysis

In this chapter some image analysis methods, which are commonly used in image processing are briefed at first. Then the methods applied to the data for this thesis are presented shortly.

3.1 Image Analysis in Other Studies

The aim of image analysis is to define methods for extracting meaningful information about the contents of a digitalized image.

In the following some methods are presented which were chosen in different studies for the analysis of similar images like our data. In these studies multiple methods are used for getting the requested results. The first step in image analysis is the process of segmentation, by which the image is splitted into its components, parts and background.

The most similar study to these thesis is a study performed in 2009 by scientists in Poland [Grala et al., 2009]. This study was about the automated image analysis methods for the assessment of Ki67 index in meningiomas (another type of brain tumors), based on the mathematical description of the cell morphology and combined with the Support Vector Machine (SVM). The materials were similarly prepared for the further image analysis, where ten microscopic areas were randomly selected and an Olympus DX-50 microscope at 400x magnification was used. The images had a resolution of 576 x 768 pixels. For

further analysis following methods had been applied: sequential thresholding, filtering and the watershed algorithm.

The algorithm they have defined, starts with the SVM tool for the separation of the immunopostitve (brown) and immunonegative (blue) cell nuclei classes. The SVM classifier of a linear kernel deliveres the appropriate value $D(x)$ for each pixel of an image that is characterized by the vector x containing three RGB- components of the pixels ([Grala et al., 2009]).
The output of the SVM, $D(x)$, which was determined for each pixel x of the original image, is used in the Sequential Thresholding Method (STM) as input described by the following equation

$$T_t(D(x)) = \begin{cases} 1 & \text{if } D(x) \geq t, \\ 0 & \text{else,} \end{cases} \tag{3.1}$$

where t is the actual bias and the STM starts from $t = \min(D(x))$ and t increases by each step until the maximum of $D(x)$ is reached. Then the watershed algorithm (see below) is used for the correction in cases of adjacent or overlapping cell nuclei. The last step is the separation into the blue and brown group. Therefore the SVM classifier of the Gaussian kernel is used. If the majority of the pixels in the cell nuclei is brown, all pixels of the nuclei are classified as immunopositive. Analogously, immunonegative nuclei are determined.

Another similar study was performed by the Department of Oncology-Pathology in Sweden, which is concerned with 2-dimensional and 3-dimensional segmentation of cell nuclei in tissue sections [Wählby et al., 2004]. A region-based method has been developed, in which seeds were created, which represent the object pixels and also the background pixels, by the combination of the morphological filtering of the original image and the gradient magnitude of the image. These seeds were used as the starting values for the watershed algorithm.
The watershed algorithm uses the intensity of an image, defined as elevation in a landscape, and splits the image into regions that are similar to the drainage regions of this landscape. The watershed borders are built at the crests in the image.
In this study the cell nuclei in the tissues were labelled with fluorescence. The images were of the size of 1024x1024 pixels and were smoothed by a 3x3 Gauss-Filter (see for more information about the Gauss-Filter [Hermes, 2005]). Then seeding process was performed using the h-extended maxima transformation (see [Wählby et al., 2004]), where all fore-

ground seeds were uniquely labelled by connected component labelling. Then the gradient magnitude image was calculated. At the local maximum of the gradient magnitude image, the seeds of the fore- and background should grow and meet, where the magnitude of the gradient expresses the variation of the local contrast in the image. Sharp edges have a high gradient magnitude, whereas the uniform areas show a gradient magnitude close to 0, and the strongest edge between the fore- and background is described by the local maximum of the gradient amplitude. Using the Sobel operators, which are a set of linear filters for the approximation of the gradients in x and y directions of the images, the gradient magnitude image was approximated. For details see [Wählby et al., 2004].

The next step in this study was the application of the watershed algorithm. The idea behind the watershed segmentation is the interpretation of the intensity image as a landscape, where every minima of the landscape is represented as a hole and the landscape is submerged in water. Then the water fills the minima and catchment basins are created. If the water rises, water from adjacent catchment basins will meet and at this points a damn (watershed) is built. These watersheds are the segmentations of the image. In the seeded watershed algorithm there are minima with pixels which are marked as seeds and unseeded local ones. The water will rise from the seeded and unseeded minima and the watersheds are built only between the catchment basins, which are associated with different seeds. The rise will stop when each seeded catchment basin in the gradient magnitude image meets another seeded catchment basin. For more information see [Wählby et al., 2004].

A study dealing with the region-based analysis about two dimensional PAGE (Polyacrylomide Gel Electrophoresis) images [Li et al., 2011], bases also on the watershed algorithm, where the algorithm was used for the segmentation of the whole gel images into regions. The aim of this study was to compare the quantities of the same protein under different treatment by comparing spot intensities.
2D PAGE is a technique which deals with separating complex mixtures, where thousands of proteins are separated and measured simultaneously.
Using the watershed algorithm, the proteins were divided into several watershed regions and the pixels in each watershed region were classified as fore- or background. Regions, which were correlated, were selected and the proteins were separated into independent sets. Then ANOVA tests were used for the independent protein regions and MANOVA tests for the correlated ones. The p-values were considered for detecting those regions with the significant changes across experimental conditions. A description of this study in detail is in [Li et al., 2011].

3.2 Preprocessing of Data

Materials for this Thesis

After the process of immunhistochemistry, see Subsection 2.4.1, the tissue sections get digitalized with a digital pathological scanner, called **NanoZoomer Digital Pathology** (NDP), which is a product suite for "Virtual Microscopy" from the Hamamatsu Corporation. During the digitalization all information of the original slide are preserved and this enables among other things a software-aided image analysis. For further information about NDP see [Hamamatsu, 2012].

Now the virtualized slides have an NDPI-format, which is based on TIFF (Tagged Image File Format) or on JPEG (Joint Photographics Expert Group) format and is the standard format of the NanoZoomer.
For further image analysis of the NDPI-data a special software development kit (SDK), called NDP.read from Hamamatsu, is needed. This enables the readout of the data.

The digitalized data were at first preprocessed by DI Andreas Walser during his masterthesis [Walser, 2011], where he has used methods of the image analysis to filter the information about the Ki67 labelled cell nuclei.
For this purpose, the development environment Microsoft Visual Studio C++ in combination with Qt, a cross-platform toolkit enabling the run and compiling of applications on several platforms such as Windows, Mac OS X, Linux, and OpenCV, a free image processing software, were used. For more information see [Walser, 2011].

The following subsections give a short overview about the used image analysis methods, separating the Ki67 labelled cell nuclei from those which are unlabelled.
The main difference between the Ki67 labelled (brown) and unlabelled (blue) cell nuclei is in the color difference. Therefore the first step is the separation of the color channels of the slide. But initially the digitalized slide has to be divided into several sectors, because the slides have on average resolutions of the size 100.000x100.000 pixels. Thus, i.e. the processing of such an image with a color depth of 24 Bit would need a memory consumption of $(10000^2 \cdot 24)/8 = 30$ Gigabyte. But the images could also have larger sizes and this would yield to the need of huge memory consumptions, i.e. 100GB or larger. Hence, a special computer would be needed.
Therefore the slides were divided into sectors of size 5000x5000 pixels. This means 75 Megabyte memory consumption per sector at a color depth of 24 Bit, and this can be used

with a usual processor with at least 4 Gigabyte computer memory. [Walser, 2011]

Methods for detecting the Ki67 labelled cell nuclei

At first the sectors with the size of 5000x5000 pixels undergo the process of separating the BGR-color channels. The BGR-color model is the same like the RGB-model, only the order of the color channels are changed. The RGB-color model consists of the overlapping of the three color channels red, green, blue, and the intensity can take a value within the interval $[0, 255]$. Through addition of the several red-, green- and blue-color shades, over several millions of color shades can be produced.
The process of color separation splits the image with a color depth of 24 Bit into 3 gray-value images with a depth of 8 Bit, where the first gray-value image consists of the intensities of blue shades, the second one of intensities of green shades and the third one of the intensities of the red shades. For the determination of the brown cell nuclei, the blue color channel seems to be the most suitable, since the blue shades do not appear, or they appear only with a small intensity within the color brown. Therefore the brown pixels appear black within the gray-value image of the blue channel and the blue ones get clear due to its intensity.

The next step is the application of the thresholding method, which enables the separation of objects from the background. There are several methods for determination of the threshold value, like the one of Nobuyuki Otsu [Otsu, 1979]. This method splits the gray values into 2 groups: the one with values bigger than the threshold value t and the one where the gray values are lower than the threshold value t. The threshold value is defined as the value where the variance between the groups reaches a maximum and the variance within the groups is minimal:

$$t = argmax\Big(\frac{Var_B(t)}{Var_W(t)}\Big). \tag{3.2}$$

For more information about the thresholding methods and the several threshold value types, see [Walser, 2011] and [Otsu, 1979]. The output image is a binary image. Pixels which belong to the background get the value 0 and pixels which belong to the foreground the value 1:

$$Im_{out}(g) = \begin{cases} 0, & \text{if } g < t, \\ 1, & \text{if } g \geq t, \end{cases} \tag{3.3}$$

where g is the gray value of the considered pixel.

The third step is to find the contours of the brown cell nuclei. Therefore the method of edge tracing is used. Binary images have two kinds of contours, internal and exterior contours. Starting with an initial value a tracing method along the contour from Satoshi Suzuki and Keiiche Abe, [Gonzales and Woods, 2008], is applied and the coordinates of the internal and exterior contours are defined.
Finally a highpass filter is applied to the binary image to distinguish the brown cell nuclei from artefacts, which developed during the labeling process. These stains are smaller and more edged than the cell nuclei. [Walser, 2011]

Methods for filtering the unlabelled cell nuclei

Analogously to the previous subsection, the unlabelled cell nuclei, which have the color blue are determined. But instead of the separation of the color channels, the conversion of the BGR color model into the HSV color model is used. The HSV stands for *Hue, Saturation and Value* and the colortype is determined through its wavelength. The wavelengths are mapped onto a ring and classified on this ring into color degrees between $0 - 360°$. Thus red has a degree of $0°$, green has $120°$ and blue has a degree of $240°$.
For the saturation two values are possible: 0 stands for a total unsaturated color and 1 for a total saturated color. The value of a color means the intensity of a color, where 0 means black and 1 the maximal possible intensity.
The conversion from the BGR color model into the HSV model is explained in detail in [Walser, 2011].

After the conversion a filter is applied, which allows the through-passing for only a specified color spectrum. The ouput is again a binary image, where the blue pixels belong to the foreground and get the value 1 and the other pixels belong to the background with the value 0. Again edge tracing is used for the determination of the contours of the blue cell nuclei, see [Walser, 2011].

Chapter 4

Mathematical Background

4.1 Distributions

Here, an overview about two important distributions, the uniform and normal distribution is given. For more information, see [Groß, 2004] and [Kütting and Sauer, 2011] and [Cramer and Kamps, 2008] and [Soong, 2004].

4.1.1 The Uniform Distribution

Definition 4.1.1 *Let $a, b \in \mathbb{R}$ with $a < b$ and $[a, b]$ be a closed interval in $\mathbb{R}$. A continuous random variable X is said to follow a uniform distribution, if it has the following probability density function $f : \mathbb{R} \to \mathbb{R}$*

$$f(x) = \begin{cases} 0, & \text{for } x < a \\ \frac{1}{b-a}, & \text{for } x \in [a, b] \\ 0, & \text{for } x > b. \end{cases} \tag{4.1}$$

The distribution function of the uniform distribution is defined as

$$F(x) = \begin{cases} 0, & \text{for } x < a \\ \frac{1}{b-a}(x-a), & \text{for } x \in [a, b] \\ 1, & \text{for } x > b. \end{cases}$$

Expectation and Variance

A uniformly distributed random variable $X \sim U([a,b])$ with the density function (4.1) has the following expectation

$$\begin{aligned}
\mu &= E(X) \\
&= \int_{-\infty}^{\infty} x \cdot f(x)dx = \int_a^b x \cdot f(x)dx \\
&= \frac{1}{b-a}\int_a^b x dx = \frac{1}{b-a}\left[\frac{x^2}{2}\right]\Big|_a^b \\
&= \frac{1}{b-a}\cdot\frac{1}{2}\cdot(b^2-a^2) = \frac{a+b}{2},
\end{aligned} \tag{4.2}$$

and the variance

$$\begin{aligned}
\sigma^2 &= Var(X) \\
&= \int_{-\infty}^{\infty}(x-\mu)^2 \cdot f(x)dx = \frac{1}{b-a}\int_a^b (x-\mu)^2 dx \\
&= \frac{1}{b-a}\int_a^b (x^2 - 2\mu x + \mu^2)dx \\
&= \frac{1}{b-a}\left[\frac{1}{3}x^3 - \mu x^2 + \mu^2 x\right]\Big|_a^b \\
&\overset{\mu=\frac{a+b}{2}}{=} \frac{1}{b-a}\left[\frac{1}{3}(b^3-a^3) - \frac{1}{2}(a+b)(b^2-a^2) + \frac{1}{4}(a+b)^2(b-a)\right]
\end{aligned}$$

$$= \frac{1}{b-a}\left[\frac{1}{12}(b^3 - a^3) + \frac{1}{4}(a^2 b - ab^2)\right]$$

$$= \frac{1}{12}\left[b^2 + ab + a^2 - 3ab\right]$$

$$= \frac{(b-a)^2}{12}. \tag{4.3}$$

Bivariate Uniform Distribution

Definition 4.1.2 *Let (X, Y) be a two dimensional random variable, where $X \sim U([a_1, b_1])$ and $Y \sim U([a_2, b_2])$ with the functional parameters $a_1 < b_1$ and $a_2 < b_2$ with $a_1, a_2, b_1, b_2 \in \mathbb{R}$ and let X and Y independent, then the joint probability density function is defined as*

$$f(x) = \begin{cases} \frac{1}{(b_1-a_1)(b_2-a_2)}, & \text{for } a_1 \leq x \leq b_1 \text{ and } a_2 \leq y \leq b_2 \\ 0, & \text{else.} \end{cases} \tag{4.4}$$

If the variables X and Y are not independent, the simple form for the bivariate density is not given. In the extreme case, if X and Y are perfectly correlated, the bivariate probability density function (pdf) has the form of a line over the (x, y)-plane.

4.1.2 The Normal Distribution

Definition 4.1.3 *A continuous random variable X is said to follow a normal distribution, if it has a probability density function $f\colon \mathbb{R} \to \mathbb{R}, x \mapsto f(x)$ with*

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

where $-\infty < \mu < \infty$ and $0 < \sigma$ are parameters describing the expected value and the standard deviation.

The distribution function of the normal distribution is defined as:

$$F(x) = \frac{1}{\sigma\sqrt{2\pi}} \int_{-\infty}^{x} \exp\left(-\frac{1}{2}\left(\frac{t-\mu}{\sigma}\right)^2\right) dt.$$

Properties of the Density Function of the Normal Distribution

- The pdf f shows a symmetric behaviour at $x = \mu$ and has a *bell shape*. For $x \geq 0 \in \mathbb{R}$ then one has $f(\mu - x) = f(\mu + x)$.

- The maximal value of f is achieved at the value $x = \mu$, and is $f(\mu) = \frac{1}{\sigma\sqrt{2\pi}}$.

- The inflection points of the pdf of the normal distribution are achieved at the values $x_1 = \mu - \sigma$ and $x_2 = \mu + \sigma$ and the functional value at these spots is $f(\mu \pm \sigma) = \frac{1}{\sigma\sqrt{2\pi}} \exp\left(-\frac{1}{2}\right)$.

The shape of the density function is affected by the parameters μ and σ, where the first one is called *location paramater* of the normal distribution and is responsible for the translations on the x-axis. The second parameter σ is the *scaling parameter* of the normal distribution and is responsible for the y-axis, where it makes the density function flatter or higher around μ.
Thus the greater the value for μ is, the righter the pdf-curve will be shifted and the greater the deviation σ is the smaller will be the maximum.

Expectation and Variance

The expectation and the standard deviation of a normally distributed random variable $X \sim N(\mu, \sigma^2)$ are equal to the parameters μ and σ:

$$
\begin{aligned}
E(X) &= \int_{-\infty}^{\infty} x \frac{1}{\sigma\sqrt{2\pi}} \exp\left(-\frac{1}{2}\left(\frac{x-\mu}{\sigma}\right)^2\right) dx \\
&\overset{y=\frac{x-\mu}{\sigma}}{=} \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{\infty} (\sigma y + \mu) \exp(-\frac{y^2}{2}) dy \\
&= \frac{\sigma}{\sqrt{2\pi}} \int_{-\infty}^{\infty} y \exp\left(-\frac{y^2}{2}\right) dy + \mu \int_{-\infty}^{\infty} \underbrace{\frac{1}{\sqrt{2\pi}} \exp\left(-\frac{y^2}{2}\right)}_{\text{Density of N(0,1)}} dy \qquad (4.5) \\
&= \frac{\sigma}{\sqrt{2\pi}} \underbrace{\left(-\exp\left(-\frac{y^2}{2}\right)\right)\Big|_{-\infty}^{\infty}}_{=0} + \mu \\
&= \mu
\end{aligned}
$$

and

$$
\begin{aligned}
E(X^2) &= \int_{-\infty}^{\infty} x^2 \frac{1}{\sigma\sqrt{2\pi}} \exp\left(-\frac{1}{2}\left(\frac{x-\mu}{\sigma}\right)^2\right) dx \\
&= \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{\infty} \frac{1}{\sigma} x^2 \exp\left(-\frac{1}{2}\left(\frac{x-\mu}{\sigma}\right)^2\right) dx \\
&\overset{y=\frac{x-\mu}{\sigma}}{=} \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{\infty} (\sigma^2 y^2 + 2\sigma y\mu + \mu^2) \exp\left(-\frac{y^2}{2}\right) dy \\
&= \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{\infty} \sigma^2 y^2 \exp\left(-\frac{y^2}{2}\right) dy + \frac{2\sigma\mu}{\sqrt{2\pi}} \underbrace{\int_{-\infty}^{\infty} y \exp\left(-\frac{y^2}{2}\right) dy}_{=0,\text{see (4.5)}} \\
&+ \mu^2 \underbrace{\int_{-\infty}^{\infty} \frac{1}{\sqrt{2\pi}} \exp\left(-\frac{y^2}{2}\right) dy}_{=1,\text{since Density of N(0,1)}}
\end{aligned}
$$

$$= \mu^2 + \frac{\sigma^2}{\sqrt{2\pi}} \int_{-\infty}^{\infty} y|y \exp\left(-\frac{y^2}{2}\right) dy|$$

$$\overset{\text{part. integr.}}{=} \mu^2 + \frac{\sigma^2}{\sqrt{2\pi}} \left[\underbrace{-\exp\left(-\frac{y^2}{2}\right) y\Big|_{-\infty}^{\infty}}_{=0} + \int_{-\infty}^{\infty} \exp\left(-\frac{y^2}{2}\right) dy \right]$$

$$= \mu^2 + \sigma^2 \underbrace{\int_{-\infty}^{\infty} \frac{1}{\sqrt{2\pi}} \exp\left(-\frac{y^2}{2}\right) dy}_{=1, \text{ since Density of N(0,1)}}$$

$$= \mu^2 + \sigma^2, \tag{4.6}$$

yields the following result for the variance $Var(X) = E(X^2) - E(X)^2 = \sigma^2$. Thus the two parameters are adequate to specify the normal distribution.

Standard Normal Distribution

Definition 4.1.4 *A random variable X with the expectation $E(X) = 0$ and the variance $Var(X) = 1$ is standard normally distributed with the following pdf $\phi(x)$*

$$\phi(x) = \frac{1}{\sqrt{2\pi}} \exp(-\frac{1}{2}x^2)$$

and the cummulative distribution function

$$\Phi(x) = \int_{-\infty}^{x} \phi(t)dt = \int_{-\infty}^{x} \frac{1}{\sqrt{2\pi}} \exp(-\frac{1}{2}t^2)dt.$$

Standardization

Since the normal distribution is entirely specified by its location and scaling parameters a linear transformation of a normally distributed random variable does not alter the type of distribution. This changes only the location and scaling.

Theorem 4.1.1 *If $X \sim N(\mu, \sigma^2)$ then one has for any scalar $a > 0$ and $b > 0$, the random variable $Z = a \pm bX$ follows a $N(a \pm b\mu, b^2\sigma^2)$ distribution.*

Proof of Theorem 4.1.1 $X \sim N(\mu, \sigma^2)$ and $Z = a + bX$ with $a > 0$ and $b > 0$, then because of the additivity and linearity of the expectation the expectation of Z is as follows

$$E(Z) = E(a + bX) = E(a) + E(bX) = a + bE(X) = a + b\mu.$$

Using (4.6) additionally the variance of Z is:

$$\begin{aligned} Var(Z) &= E(Z^2) - (E(Z))^2 \\ &= E((a + bX)^2) - \underbrace{(E(a + bX))^2}_{(a+b\mu)^2} \\ &= a^2 + 2ab\mu + b^2 \underbrace{E(X^2)}_{\mu^2+\sigma^2} - a^2 - 2ab\mu - b^2\mu^2 \\ &= b^2\sigma^2. \end{aligned}$$

Thus a normally distributed random variable is invariant to linear transformation. □

If X is a random variable with $X \sim N(\mu, \sigma^2)$ then the random variable $Z = \frac{X-\mu}{\sigma}$ is called standardized and normally distributed with the parameters $E(Z) = 0$ and $Var(Z) = 1$.

The relationship between pdf $f(x)$ of the $N(\mu, \sigma^2)$ and the pdf $\phi(x)$ of the $N(0, 1)$ is shown through

$$f(x) = \frac{1}{\sigma}\phi\left(\frac{x - \mu}{\sigma}\right),$$

and for the cummulative distribution functions one has following relation:

Theorem 4.1.2 *If $X \sim N(\mu, \sigma^2)$ then the cdf $F(x)$ of X is given by*

$$F(x) = \Phi\left(\frac{x - \mu}{\sigma}\right)$$

for $-\infty < x < \infty$.

There is a proof of Theorem 4.1.2 for instance on p. 313 in [Kütting and Sauer, 2011].

Bivariate Normal Distribution

Definition 4.1.5 *Let (X, Y) be a two-dimensional random variable having a bivariate normal distribution with parameters μ_1, μ_2, $\sigma_1 > 0$, $\sigma_2 > 0$ and $-1 < \rho < 1$ then one has the following joint pdf of X and Y:*

$$f(x,y) = \frac{1}{2\pi\sigma_1\sigma_2\sqrt{1-\rho^2}} \exp\left(-\frac{1}{2}Q(x,y;\mu_1,\mu_2,\sigma_1,\sigma_2,\rho)\right), \tag{4.7}$$

where

$$Q(x,y;\mu_1,\mu_2,\sigma_1,\sigma_2,\rho) = \frac{1}{(1-\rho^2)}\left[\frac{(x-\mu_1)^2}{\sigma_1^2} - 2\rho\frac{(x-\mu_1)(y-\mu_2)}{\sigma_1\sigma_2} + \frac{(y-\mu_2)^2}{\sigma_2^2}\right]. \tag{4.8}$$

Contours of the bivariate normal distribution are defined as follows:

Definition 4.1.6 *If the parameters μ_1, μ_2, σ_1, σ_2 and ρ are given and $c \in \mathbb{R}$ is fix, then the bivariate normal pdf $f(x,y)$ is the same for any point (x,y) from the set of points*

$$\{(x,y) : Q(x,y;\mu_1,\mu_2,\sigma_1,\sigma_2,\rho) = c^2\}. \tag{4.9}$$

For different values c^2 is the set above called contours of the bivariate normal distribution and they describe in the xy-plane ellipses with the center (μ_1, μ_2).

Special cases:

- *If $\rho = 0$ then the axes of the ellipses are parallel to the x- and y-axis.*
- *If $\rho = 0$ and $\sigma_1 = \sigma_2$ the contours describe circles in the xy-plane, since both axes have the same length.*

In Figure 4.1 there is an example for the probability density function of the bivariate normal distribution and also a respective contour plot.

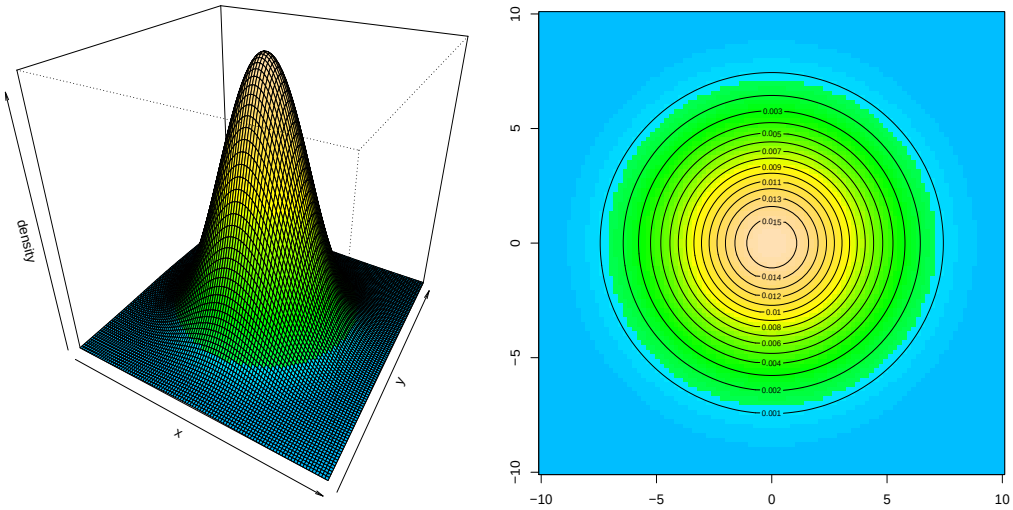

Figure 4.1: Probability density function of the bivariate normal distribution with parameters $\mu_1 = \mu_1 = 0$ und $\sigma_1 = \sigma_2 = 10$, where the correlation is $\rho = 0$.

If the case where $\rho = 0$ and $\sigma_1 = \sigma_2 = \sigma$ and $\mu_1 = \mu_2 = 0$ is considered, the pdf of the bivariate normal distribution is

$$f(x, y) = \frac{1}{2\pi\sigma^2} \exp\left(-\frac{1}{2}\left(\frac{x^2}{\sigma^2} + \frac{y^2}{\sigma^2}\right)\right) \tag{4.10}$$

and the contours of the bivariate normal distribution can be calculated analytically in the following way:
Let $k \in \mathbb{R}$ be a constant value, then the contour describes the place where $f(x, y) = k$. Thus one has

$$\frac{1}{2\pi\sigma^2} \exp\left(-\frac{1}{2}\left(\frac{x^2}{\sigma^2} + \frac{y^2}{\sigma^2}\right)\right) = k$$

$$\exp\left(-\frac{1}{2}\left(\frac{x^2}{\sigma^2} + \frac{y^2}{\sigma^2}\right)\right) = 2\pi\sigma^2 k$$

$$-\frac{1}{2}\Big(\frac{x^2}{\sigma^2}+\frac{y^2}{\sigma^2}\Big) = \ln(2\pi\sigma^2 k)$$

$$\frac{x^2}{\sigma^2}+\frac{y^2}{\sigma^2} = -2\ln(2\pi\sigma^2 k)$$

$$x^2+y^2 = -2\sigma^2\ln(2\pi\sigma^2 k). \tag{4.11}$$

Equation (4.11) describes a circle with the center $M=(0,0)$ and the radius $r=\sqrt{-2\sigma^2\ln(2\pi\sigma^2 k)}$. For $\mu_1 \neq \mu_2 \neq 0$ the center of the circle is shifted to $M=(\mu_1,\mu_2)$.

If $\rho = 0$ and $\sigma_1 > \sigma_2$ then the contours describe ellipses of the form

$$\frac{(x-\mu_1)^2}{2\sigma_1^2\ln(\frac{1}{2\pi\sigma_1\sigma_2 k})}+\frac{(y-\mu_2)^2}{2\sigma_2^2\ln(\frac{1}{2\pi\sigma_1\sigma_2 k})}=1,$$

with center (μ_1,μ_2) and axis lengths $2\cdot\sqrt{2\sigma_1^2\ln(\frac{1}{2\pi\sigma_1\sigma_2 k})}$ and $2\cdot\sqrt{2\sigma_2^2\ln(\frac{1}{2\pi\sigma_1\sigma_2 k})}$.

If the correlation $\rho \neq 0$, then the contours are tilted ellipses with center (μ_1,μ_2) .

Properties of the Bivariate Normal Distribution

Theorem 4.1.3 *If the random variable (X,Y) has a bivariate normal distribution with the parameters μ_1, μ_2, $\sigma_1>0$, $\sigma_2>0$ and $-1<\rho<1$ then:*

(i) $X \sim N(\mu_1,\sigma_1^2)$,

(ii) $Y \sim N(\mu_2,\sigma_2^2)$.

Notice: The converse of statements (i) and (ii) from the Theorem above is not true in general, but only in the case that X and Y are independent (see [Groß, 2004] p. 177).

Proof of Theorem 4.1.3 :
$(X,Y) \sim N(\mu_1,\mu_2,\sigma_1^2,\sigma_2^2,\rho)$ with the pdf in (4.7) and if one defines

$$\tilde{x}=\frac{x-\mu_1}{\sigma_1} \quad and \quad \tilde{y}=\frac{y-\mu_2}{\sigma_2},$$

then one has for the bivariate pdf

$$f(x,y) = \frac{1}{2\pi\sigma_1\sigma_2\sqrt{1-\rho^2}} \exp\left(-\frac{1}{2(1-\rho^2)}(\tilde{x}^2 - 2\rho\tilde{x}\tilde{y} + \tilde{y}^2)\right). \tag{4.12}$$

Now if the expression in the exponent is considered and completed to a square by adding and removing $\rho^2\tilde{x}^2$, one has

$$(\tilde{x}^2 - 2\rho\tilde{x}\tilde{y} + \tilde{y}^2) = \tilde{x}^2(1-\rho^2) + (\tilde{y} - \rho\tilde{x})^2.$$

That implies for the pdf in (4.12) the following factorization:

$$f(x,y) = g(x) \cdot h(x,y)$$

with

$$\begin{aligned} g(x) &= \frac{1}{\sqrt{2\pi}\sigma_1} \exp\left(-\frac{\tilde{x}^2}{2}\right) \\ &= \frac{1}{\sqrt{2\pi}\sigma_1} \exp\left(-\frac{(x-\mu_1)^2}{2\sigma_1^2}\right) \\ &\sim N(\mu_1, \sigma_1^2), \end{aligned}$$

$$\begin{aligned} h(x,y) &= \frac{1}{\sqrt{2\pi}\sigma_2\sqrt{1-\rho^2}} \exp\left(-\frac{1}{2(1-\rho^2)}(\tilde{y} - \rho\tilde{x})^2\right) \\ &= \frac{1}{\sqrt{2\pi}\sigma_2\sqrt{1-\rho^2}} \exp\left(-\frac{1}{2(1-\rho^2)\sigma_2^2}(y - (\mu_2 + \rho\sigma_2\tilde{x}))^2\right) \\ &\sim N(\mu_2 + \rho\sigma_2\tilde{x}, (1-\rho^2)\sigma_2^2). \end{aligned}$$

The marginal density of X is

$$f_X(x) = \int_{-\infty}^{\infty} f(x,y)dy$$

$$= \int_{-\infty}^{\infty} g(x)h(x,y)dy$$

$$= g(x) \underbrace{\int_{-\infty}^{\infty} h(x,y)dy}_{=1} = g(x).$$

That implies that the marginal density of X is again a normal distribution with parameters μ_1 and σ_1^2. Analogously the marginal density of $Y \sim N(\mu_2, \sigma_2^2)$. □

Theorem 4.1.4 *If (X,Y) has a bivariate normal distribution with the parameters μ_1, μ_2, $\sigma_1 > 0$, $\sigma_2 > 0$ and $-1 < \rho < 1$ then one has for $a \neq 0$ and $b \neq 0$:*

$$aX + bY \sim N(a\mu_1 + b\mu_2, \quad a^2\sigma_1^2 + 2\rho ab\sigma_1\sigma_2 + b^2\sigma_2^2).$$

The converse of this statement is true in the case X and Y are not perfectly correlated, i.e. $|\rho_{X,Y}| < 1$.

4.2 Kernel Density Estimation

This subsection gives an overview about the subject of kernel density estimation and its properties. This topic is well discussed e.g. in [Härdle et al., 2004] and [Scott, 1992].

The idea behind kernel density estimation is to define an estimator so that the estimation of a smooth continuous probability density function is possible and which is free of the problem of choosing the origin of a bin grid.
A reasonable way to estimate the pdf $f(x)$ is to calculate

$$\frac{1}{n \cdot \text{interval length}} \#\{\text{observations that fall into a small interval around } x\}, \qquad (4.13)$$

where the considered interval is of the form $[x-h, x+h)$ and hence it has a length of $2h$ with h being the binwidth.
Thus (4.13) is equal to

$$\hat{f}_h(x) = \frac{1}{2hn} \sum_{i=1}^{n} \mathbb{I}(x_i \in [x-h, x+h]), \quad i = 1, \ldots, n. \tag{4.14}$$

Now consider a random sample $X_1, \ldots, X_n$. If the observations X_i which fall into the interval $[x-h, x+h)$ are weighted through a *kernel function* K as weighting function, (4.14) yields the following formula

$$\hat{f}_h(x) = \frac{1}{nh} \sum_{i=1}^{n} K\Big(\frac{x - X_i}{h}\Big). \tag{4.15}$$

Kernel Function

Definition 4.2.1 *Let $X_1, \ldots, X_n$ be a random sample and $u\colon X \times X \to \mathbb{R}$ a distance measure, then a mapping $K\colon \mathbb{R} \to \mathbb{R}$, $u \mapsto K(u)$ with the following properties*

- $K(u) \geq 0$,
- $\int K(u)du = 1$,

is called a kernel function. K(u) has the following regularity conditions:

(i) $K(u) = K(-u)$,

(ii) $K(u)$ *is bounded,*

(iii) $|u|K(u) \to 0$ *for* $|x| \to 0$,

(iv) $\int u^2 K(u)du < \infty$.

Examples for Kernel Functions

Uniform Kernel	$K(u) =$	$\frac{1}{2}\mathbb{I}(\lvert u\rvert \leq 1)$,
Triangle Kernel	$K(u) =$	$(1 - \lvert u\rvert \leq 1)\mathbb{I}(\lvert u\rvert \leq 1)$,
Epanechnikov Kernel	$K(u) =$	$\frac{3}{4}(1 - u^2)\mathbb{I}(\lvert u\rvert \leq 1)$,
Gaussian Kernel	$K(u) =$	$\frac{1}{\sqrt{2\pi}} \exp(-\frac{1}{2}u^2)$,

where $u = \frac{x - X_i}{h}$ is the scaled distance and $\mathbb{I}(|u| \leq 1)$ the indicator function.

Figure 4.2 displays the above mentioned kernel functions.

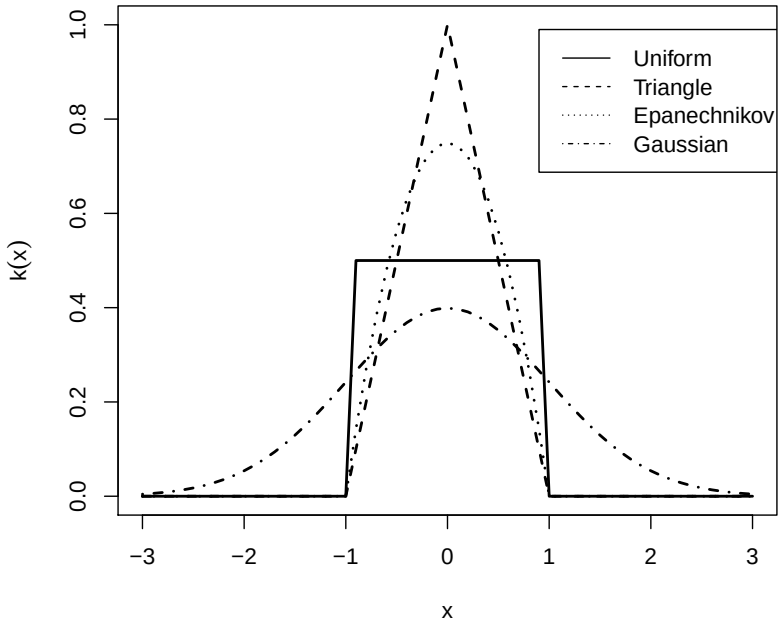

Figure 4.2: Some kernel functions

4.2.1 Univariate Kernel Density Estimation

Definition 4.2.2 *Let $X_1, \ldots, X_n$ be a random sample and $K\colon \mathbb{R} \to \mathbb{R}$ a kernel function like in Definition 4.2.1.*

$$\hat{f}_h(x) = \frac{1}{n}\sum_{i=1}^{n} K_h(x - X_i), \tag{4.16}$$

where

$$K_h(x, X_i) = \frac{1}{h}K\Big(\frac{x - X_i}{h}\Big),$$

is called univariate kernel density estimator with binwidth h and kernel K.

Since the kernel function is by definition a probability density function, this implies that the kernel density estimator is a pdf too, i.e. $\int K(u)du = 1 \Rightarrow \int \hat{f}(x)dx = 1$.
In addition, $\hat{f}$ inherits all the continuity and differentiability properties of K.

Statistical Properties

The expectation of the kernel density estimation is calculated as

$$\begin{aligned} E(\hat{f}_h(x)) &= E\Big(\frac{1}{n}\sum_{i=1}^{n} K_h(x - X_i)\Big) \\ &= \frac{1}{n}\sum_{i=1}^{n} E(K_h(x - X_i)) \\ &= E(K_h(x - X_i)) \\ &= \frac{1}{h}\int K\Big(\frac{x - u}{h}\Big) f(u)du \end{aligned} \tag{4.17}$$

and the variance as

$$\begin{aligned} Var(\hat{f}_h(x)) &= Var\Big(\frac{1}{n}\sum_{i=1}^{n} K_h(x - X_i)\Big) \\ &= \frac{1}{n^2}\sum_{i=1}^{n} Var(K_h(x - X_i)) \\ &= \frac{1}{n} Var(K_h(x - X_i)) \\ &= \frac{1}{n}\Big(E(K_h(x - X_i))^2 - [E(K_h(x - X_i))]^2\Big), \end{aligned} \tag{4.18}$$

where

$$\frac{1}{n}(E(K_h(x - X_i))^2) = \frac{1}{nh^2}\int \Big(K\Big(\frac{x-u}{h}\Big)\Big)^2 f(u)du. \tag{4.19}$$

In addition, for the Bias the following calculations can be done:

$$\begin{aligned} Bias(\hat{f}_h(x)) &= E(\hat{f}_h(x)) - f(x) \\ &\overset{\text{by(4.17)}}{=} \frac{1}{h}\int K\Big(\frac{x-u}{h}\Big) f(u)du - f(x). \end{aligned} \tag{4.20}$$

Now using the regularity conditions of the kernel function (see 4.2.1) and defining $s = \frac{u-x}{h}$ and a second order Taylor expansion of f(u) around x: $f(x + sh) = f(x) - hsf'(x) + \frac{h^2 s^2}{2} f''(x) + o(h^2)$ yields the following result

$$Bias(\hat{f}_h(x)) = \frac{1}{h}\int K\Big(\frac{x-u}{h}\Big)f(u)du - f(x)$$

$$= -hf'(x)\underbrace{\int sK(s)ds}_{=0} + \frac{h^2 f''(x)}{2}\underbrace{\int s^2 K(s)ds}_{=:\mu_2(K)} + o(h^2)$$

$$= \frac{h^2 f''(x)}{2}\mu_2(K) + o(h^2), \text{ as } h \to 0. \tag{4.21}$$

Hence the bias is proportional to h^2 and therefore a small h reduces the bias. Furthermore it depends on the $f''(x)$ and large values of $|f''(x)|$ imply large values of the bias of the kernel density estimator.

Analogously for the variance performing a second taylor expansion with similar variable substitution and using Equation (4.19) and $E(K_h(x-X)) = f(x) + o(h)$ yields

$$Var(\hat{f}_h(x)) = \frac{1}{nh}\underbrace{\int K(s)^2 ds}_{||K||_2^2} f(x) + o\Big(\frac{1}{nh}\Big), \text{ as } nh \to \infty. \tag{4.22}$$

Since the variance of the kernel density estimator is nearly proportional to $\frac{1}{nh}$ a very large value for h is needed to keep the variance small.
The aim is to keep the variance and the bias small. But increasing h will lower the variance while it will raise the bias and decreasing will do the opposite ($\Rightarrow$ trade-off between variance and bias).

A compromise to avoid over- and undersmoothing is in minimizing the MSE, the sum between the variance and squared bias.

$$MSE(\hat{f}_h(x)) = Bias(\hat{f}_h(x))^2 + Var(\hat{f}_h(x))$$

$$= \frac{h^4}{4}f''(x)^2\mu_2(K)^2 + \frac{1}{nh}||K||_2^2 f(x) + o(h^4) + o\Big(\frac{1}{nh}\Big). \tag{4.23}$$

This shows that the MSE of the kernel density estimator goes to zero for $h \to 0$ and

$nh \to \infty$. Thus the kernel density estimator is consistent. Furthermore, the MSE depends on f and f'' and does not drop out by deriving and thus in practice, it is not applicable to derive an optimal value for h by minimizing the MSE.

Derivation of the Optimal Bandwidth

The Mean Integrated Squared Error (MISE) for the kernel density estimator is calculated as follows

$$MISE(\hat{f}_h) = \int MSE(\hat{f}_h(x))dx$$

$$\overset{\text{by } (4.23)}{=} \frac{1}{nh}||K||_2^2 \underbrace{\int f(x)dx}_{=1} + \frac{h^4}{4}\mu_2(K)^2 \int f''(x)^2 dx + o(h^4) + o\Big(\frac{1}{nh}\Big)$$

$$= \frac{1}{nh}||K||_2^2 + \frac{h^4}{4}\mu_2(K)^2||f''||_2^2 + o\Big(\frac{1}{nh}\Big) + o(h^4), \tag{4.24}$$

as $h \to 0, nh \to \infty$,

and this yields the approximated MISE (AMISE) by ignoring higher order terms

$$AMISE(\hat{f}_h) = \frac{1}{nh}||K||_2^2 + \frac{h^4}{4}\mu_2(K)^2||f''||_2^2. \tag{4.25}$$

Through differentiation the AMISE with respect to h and solving the first-order condition, the AMISE-optimal bandwith is calculated:

$$-\frac{1}{nh^2}||K||_2^2 + h^3\mu_2(K)^2||f''||_2^2 = 0$$

$$h^5\mu_2(K)^2||f''||_2^2 = \frac{1}{n}||K||_2^2$$

$$h_{opt} = \left(\frac{||K||_2^2}{||f''||_2^2 \mu_2(K)^2 n} \right)^{\frac{1}{5}} \sim n^{-\frac{1}{5}}. \tag{4.26}$$

Still, the optimal bandwidth h_{opt} depends on an unknown quantity $||f''||_2^2$. Therefore a plug-in method introduced by Silverman (see e.g. [Silverman, 1986] and [Sheather and Jones, 1991]) will be derived. The main idea is to replace the unknown parameter through an estimate. Since $||f''||_2^2$ is the unknown quantity, the assumption that f belongs to the family of normal distributions with mean μ and variance σ^2 is made and the following result is obtained:

$$\begin{aligned} ||f''||_2^2 &= \sigma^{-5} \int (\phi''(x))^2 dx \\ &= \sigma^{-5} \frac{3}{8\sqrt{\pi}} \qquad \approx 0.212\sigma^{-5}, \end{aligned} \tag{4.27}$$

where $\phi(\cdot)$ describes the pdf of the standard normal distribution.
Now the unknown σ has to be replaced by the estimator

$$\hat{\sigma} = \sqrt{\frac{1}{n-1} \sum_{i=1}^{n} (x_i - \overline{x})^2}$$

and taking the Gaussian kernel as kernel function the following optimal bandwidth is obtained

$$\begin{aligned} \hat{h}_{rot} &= \left(\frac{||\phi||_2^2}{||\hat{f}''||_2^2 \mu_2(\phi)^2 n} \right)^{\frac{1}{5}} \\ &\overset{\text{by (4.27)}}{=} \left(\frac{4\hat{\sigma}^5}{3n} \right)^{\frac{1}{5}} \qquad \approx 1.06\hat{\sigma} n^{-\frac{1}{5}}, \end{aligned} \tag{4.28}$$

where the Gaussian kernel is identical to the pdf of the standard normal distribution. This bandwidth is called the "rule-of-thumb" bandwidth.

Since the rule-of-thumb bandwidth is sensitive to outliers, a more robust estimator for σ is given through the interquartile range

$$R = X_{[0.75n]} - X_{[0.25n]}.$$

Assuming that the true pdf is normal and $X \sim N(\mu, \sigma^2)$ and $Z = \frac{X-\mu}{\sigma} \sim N(0,1)$ the following result for R is obtained

$$\begin{aligned} R &= X_{[0.75n]} - X_{[0.25n]} \\ &= (\mu + \sigma Z_{[0.75n]}) - (\mu + \sigma Z_{[0.25n]}) \\ &= \sigma(Z_{[0.75n]} - Z_{[0.25n]}) \\ &\approx \sigma(0.67 - (-0.67)) = 1.34\sigma. \end{aligned} \tag{4.29}$$

This implies for the estimator of σ

$$\hat{\sigma} = \frac{\hat{R}}{1.34},$$

where $\hat{R}$ is the estimated interquartile range.
Plugging this into the rule-of-thumb bandwidth in (4.28) yields

$$\hat{h}_{rot} = 1.06 \frac{\hat{R}}{1.34} n^{-\frac{1}{5}} \approx 0.79 \hat{R} n^{-\frac{1}{5}}. \tag{4.30}$$

The combination of (4.28) and (4.30) gives a "better rule-of-thumb" bandwidth as

$$\hat{h}_{rot} = 1.06 \min\left\{\hat{\sigma}, \frac{\hat{R}}{1.34}\right\} n^{-\frac{1}{5}}. \tag{4.31}$$

4.2.2 Bivariate Kernel Density Estimator

Analogously to the univariate kernel density estimator, the bivariate and in general a multivariate kernel density estimator can be defined. In the following the bivariate one is considered.

Definition 4.2.3 *Let $\boldsymbol{X} = (X_1, X_2)^T \in \mathbb{R}^2$ be an i.i.d random variable, then a function $K\colon \mathbb{R}^2 \to \mathbb{R}$ is called a bivariate kernel function if for $\boldsymbol{u} \in \mathbb{R}^2$*

$$\int K(\boldsymbol{u}) d\boldsymbol{u} = 1$$

is complied and it fulfills the following regularity conditions

(i) $\int \boldsymbol{u} K(\boldsymbol{u}) d\boldsymbol{u} = 0$,

(ii) $\int \boldsymbol{u}\boldsymbol{u}^T K(\boldsymbol{u}) d\boldsymbol{u} = \mu_2(K)\boldsymbol{I}_2$,

where $\boldsymbol{I}_2$ denotes the 2×2 identity matrix.

An example for a bivariate kernel function is the bivariate Gaussian kernel

$$K(\boldsymbol{u}) = \frac{1}{2\pi} \exp\Big(-\frac{1}{2}\boldsymbol{u}^T\boldsymbol{u}\Big).$$

Definition 4.2.4 *Let $K\colon \mathbb{R}^2 \to \mathbb{R}$ a kernel function and $X_i = (X_{i1}, X_{i2})^T$ a random Sample, for $i = 1, \ldots, n$. Then for any $\boldsymbol{x} = (x_1, x_2)^T$,*

$$\hat{f}_h(\boldsymbol{x}) = \frac{1}{n}\sum_{i=1}^{n} \frac{1}{h^2} K\Big(\frac{\boldsymbol{x} - \boldsymbol{X}_i}{h}\Big)$$

$$= \frac{1}{n}\sum_{i=1}^{n} \frac{1}{h^2} K\Big(\frac{x_1 - X_{i1}}{h}, \frac{x_2 - X_{i2}}{h}\Big) \tag{4.32}$$

is called a bivariate kernel density estimator with the same bandwidth h for both components and kernel K.

If a vector of bandwidths $\boldsymbol{h} = (h_1, h_2)^T$ is considered, then one has the following bivariate kernel density estimator

$$\hat{f}_h(\boldsymbol{x}) = \frac{1}{n}\sum_{i=1}^{n} \frac{1}{h_1 \cdot h_2} K\Big(\frac{x_1 - X_{i1}}{h_1}, \frac{x_2 - X_{i2}}{h_2}\Big). \tag{4.33}$$

Furthermore, the bivariate kernel function is of the form $K(\boldsymbol{u}) = K(u_1) \cdot K(u_2)$ for $\boldsymbol{u} = (u_1, u_2)^T$, and the bivariate kernel density estimator in (4.33) becomes

$$\hat{f}_h(\boldsymbol{x}) = \frac{1}{n}\sum_{i=1}^{n} \frac{1}{h_1 \cdot h_2} K\Big(\frac{x_1 - X_{i1}}{h_1}\Big) K\Big(\frac{x_2 - X_{i2}}{h_2}\Big). \tag{4.34}$$

Using a Gaussian kernel the following bivariate kernel density estimator is obtained

$$\hat{f}_h(\boldsymbol{x}) = \frac{1}{n}\sum_{i=1}^{n} \frac{1}{2\pi h_1 h_2} \exp\left(-\frac{1}{2}\left(\frac{x_1 - X_{i1}}{h_1}\right)^2\right) \cdot \exp\left(-\frac{1}{2}\left(\frac{x_2 - X_{i2}}{h_2}\right)^2\right), \tag{4.35}$$

for $X_{i1} \in [x_1 - h_1, x_1 + h_1)$ and $X_{i2} \in [x_2 - h_2, x_2 + h_2)$.

Statistical Properties

Analogously to the univariate case, the expectation, variance and bias can be calculated by using a second order Taylor expansion of f around $\boldsymbol{x}$:

$$f(\boldsymbol{x} + \boldsymbol{t}) = f(\boldsymbol{x}) + \boldsymbol{t}^T \nabla_f(\boldsymbol{x}) + \frac{1}{2}\boldsymbol{t}^T \mathcal{H}_f(\boldsymbol{x})\boldsymbol{t} + o(\boldsymbol{t}^T\boldsymbol{t}), \tag{4.36}$$

where $\nabla_f(\boldsymbol{x}) = (\frac{\partial f}{\partial x_1}, \frac{\partial f}{\partial x_2})^T$ is the gradient and $\mathcal{H}_f(\mathbf{x}) = \begin{pmatrix} \frac{\partial^2 f}{\partial x_1 \partial x_1} & \frac{\partial^2 f}{\partial x_1 \partial x_2} \\ \frac{\partial^2 f}{\partial x_2 \partial x_1} & \frac{\partial^2 f}{\partial x_2 \partial x_2} \end{pmatrix}$ the Hessian matrix of f, and $\boldsymbol{t} \in \mathbb{R}^2$.

The expectation is obtained through

$$\begin{aligned} E(\hat{f}_{\mathbf{H}}(\boldsymbol{x})) &= \int K_{\mathbf{H}}(\boldsymbol{u}-\boldsymbol{x}) f(\boldsymbol{u}) d\boldsymbol{u} \\ &= \int K(\boldsymbol{s}) f(\boldsymbol{x}+\mathbf{H}\boldsymbol{s}) d\boldsymbol{s} \\ &\approx \int K(\boldsymbol{s})[f(\boldsymbol{x}) + \boldsymbol{s}^T\mathbf{H}^T\nabla_f(\boldsymbol{x}) + \frac{1}{2}\boldsymbol{s}^T\mathbf{H}^T\mathcal{H}_f(\boldsymbol{x})\mathbf{H}\boldsymbol{s}] d\boldsymbol{s}, \end{aligned} \tag{4.37}$$

where $\mathbf{H} = diag(h_1, h_2)$ is a diagonal matrix of the bandwidths.

For the variance one has

$$\begin{aligned} Var(\hat{f}_{\mathbf{H}}(\boldsymbol{x})) &= \frac{1}{n}\int (K_{\mathbf{H}}(\boldsymbol{u}-\boldsymbol{x}))^2 d\boldsymbol{u} - \frac{1}{n}(E(\hat{f}_{\mathbf{H}}(\boldsymbol{x})))^2 \\ &\approx \int \frac{1}{nh_1h_2} K(\boldsymbol{s})^2 f(\boldsymbol{x}+\mathbf{H}\boldsymbol{s}) d\boldsymbol{s} \\ &\approx \int \frac{1}{nh_1h_2} K(\boldsymbol{s})^2 \Big(f(\boldsymbol{x}) + \boldsymbol{s}^T\mathbf{H}^T\nabla_f(\boldsymbol{x}) \Big) d\boldsymbol{s} \\ &\approx \frac{1}{nh_1h_2} ||K||_2^2 f(\boldsymbol{x}). \end{aligned} \tag{4.38}$$

Furthermore, the regularity conditions in Definition 4.2.3 and (4.37) yields $E(\hat{f}_{\mathbf{H}}(\boldsymbol{x})) - f(\boldsymbol{x}) \approx \frac{1}{2}\mu_2(K)\ \mathrm{tr}(\mathbf{H}^T\mathcal{H}_f(\boldsymbol{x})\mathbf{H})$, and now the bias is obtained by

$$Bias(\hat{f}_{\mathbf{H}}(\boldsymbol{x})) \approx \frac{1}{4}\mu_2^2(K) \int \Big(\ \mathrm{tr}(\mathbf{H}^T\mathcal{H}_f(\boldsymbol{x})\mathbf{H})\Big)^2 d\boldsymbol{x}, \tag{4.39}$$

where tr is the trace of a matrix.

There are several methods for the determination of the matrix of the bandwidths in [Härdle et al., 2004] and [Scott, 1992]. In this thesis those are calculated like in Equation (4.31).

4.3 Distance Measures

In the following the term of "distance" is defined, see [Schlittgen, 2009].

Definition 4.3.1 *Let X be a random set. The mapping $d\colon X \times X \to \mathbb{R}$ is said to be a metric, if it fulfills the following conditions:*
$\forall x, y, z \in X$:

- $d(x, x) = 0$,
- $d(x, y) > 0$ *if* $x \neq y$,
- $d(x, y) = d(y, x)$,
- $d(x, y) \leq d(x, z) + d(z, y)$.

An important group of metrices is constituted by the L_p - distances, also called *Minkowski*-metrices, and which is defined for the measurements $x_1, \ldots, x_n$ of the quantity x and $y_1, \ldots, y_n$ of the quantity y as:

$$d_p(x, y) = \left(\sum_{i=1}^{q} |x_i - y_i|^p \right)^{\frac{1}{p}}, \quad p \geq 1. \tag{4.40}$$

Definition 4.3.2 *For the case $p = 1$ the metric in* (4.40) *yields the L_1-distance, which is defined as*

$$d_1(x, y) = \sum_{i=1}^{n} |x_i - y_i|. \tag{4.41}$$

If $p = 2$ the distance in (4.40) *gives the Euclidean-distance, defined as*

$$d_2(x, y) = \left(\sum_{i=1}^{n} (x_i - y_i)^2 \right)^{\frac{1}{2}}.$$

Chapter 5

Definition of Indicators

In this chapter the aim is to define appropriate measures, which deliver information about the behaviour of the distribution of measurements analyzed in Chapter 6.
Both first indicators are measures for the inequality within the behavior of the distribution. The last two will give information on the spatial distribution of possible groupings.

5.1 Indicator 1: giniUTR

The goal is to consider the behaviour of the density estimation by considering points of intersection with the density along the z-axis. Thus the first step is the definition of the ratio 5.1.

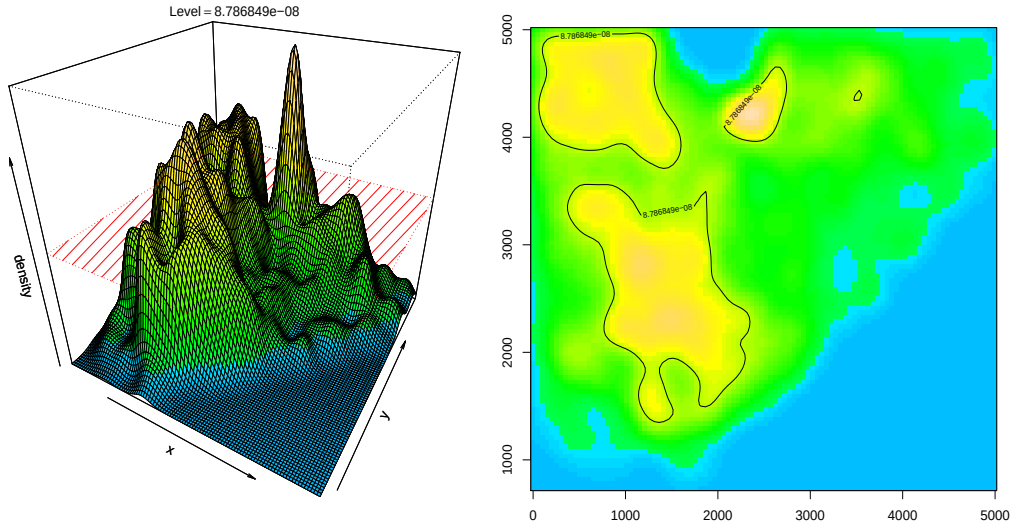

Figure 5.1: KDE with a level cut along the z-axis.

Figure 5.1 presents a plot of an example for a kernel density estimation and a level cut along the z-axis, where the vector of level cuts is given through $\boldsymbol{\alpha} \in \mathbb{R}^m$. The plot on the right side shows the contours of the density estimation, where the yellow areas represent the values at the level cut $\alpha_3 = 8.786849e - 08$ and the green area, including the yellow areas too, describes the area at the first level α_1. The following ratio is now supposed to set the yellow areas in relation to the green one.

Upper-To-Total - Ratio

At first the **upper-to-total**- ratio (UTR) is calculated. Suppose we have given a sample $x_1, \ldots, x_n$. For this purpose a vector $\boldsymbol{\alpha} = (\alpha_1, \ldots, \alpha_m)^T \in \mathbb{R}^m$ is defined, which describes the level cuts along the z-axis of the bivariate density estimation. At each level cut α_j, $j \in \{1, \ldots, m\}$, the ratio of the number of the values of the density estimation that are bigger than the level α_j is set in relation to the total number of the values of the density estimation, where α_1 is the level at the baseline. Since the density estimation contains values too close to zero, a baseline, as the minimal level of all levels, is the defined and only values larger than this baseline are considered for the analysis:

$$UTR_j = \frac{\sum_{l=1}^{N} \mathbb{I}(y_l \geq \alpha_j)}{\sum_{l=1}^{N} \mathbb{I}(y_l \geq \alpha_1)}, \tag{5.1}$$

with

$$\mathbb{I}(y_l \geq \alpha_j) = \begin{cases} 1, & \text{for } y_l \geq \alpha_j \\ 0, & \text{else ,} \end{cases}$$

where $UTR_j \in [0,1]$, $\alpha_j \in \mathbb{R}$ and y_l with $l = 1, \ldots, N$ are the values of the density estimation.

Since the UTR is considered for several level cuts α_j, a vector of ratio-values in descending order is obtained and thus UTR is a m-dimensional vector of values, where each component is in the interval $[0,1]$.

Gini-Index

The Gini-Coefficient, also known as Gini-Index, is a measure for concentration respectively a measure of inequality considering all parts of the distribution. It enables to compare the inequality of two groups or data directly independent of their size. There are different ways for the calculation of the Gini-Index. The algebraic way is to compute directly from the following formula, if the data $\boldsymbol{x} = (x_1, \cdots, x_n)^T$ is ordered from the smallest to the largest value [Travis, 2008]:

$$gini(\boldsymbol{x}) = \frac{\sum_{i=1}^{n}(2i - n - 1)x_i}{n^2\mu} \qquad \in [0,1], \tag{5.2}$$

where i describes the rank order number and $\mu = \frac{1}{n}\sum_{i=1}^{n} x_i$.

If the Gini-Index takes the value 0 then there is a perfect equality within the data. There is a perfect inequality resp. a concentration to one value if the Gini-Index is 1. Hence the smaller the Gini is, the closer the data are to equality.

For the graphical interpretation of the Gini-Index, the Lorenz-Curve is considered. The Lorenz-Curve describes the relative concentration, where the observations resp. considered data is ordered from the lowest to the highest value. Then the cummulative proportion of the relative frequencies (u) is calculated for the x-axis and the cummulative proportion of the variables of interest (v) for the y-axis,

$$u = \frac{j}{n}$$

and

$$v = \frac{\sum_{i=1}^{j} x_i}{\sum_{i=1}^{n} x_i}.$$

Example

5 companies are considered, with the following sales (see [Faes, 2007]):

Company	1	2	3	4	5
Sales in Million €	20	50	15	15	20

Table 5.1: Example: Sales of 5 companies

Then the companies are sorted with regard to their sale and the computation of the cummulative proportion of the relative frequencies (u) and the cummulative proportion of the variables of interest (v) result in:

Company	3	4	1	5	2
Sales in Million €	15	15	20	20	50
u	0.2	0.4	0.6	0.8	1.0
v	0.125	0.250	0.417	0.583	1.000

Table 5.2: Example: Cummulative proportion of the relative frequencies (u) and the cummulative proportion of the variables of interest (v)

In Figure 5.2 the obtained results for u and v are plotted, where the equality line is plotted as a diagonal line.

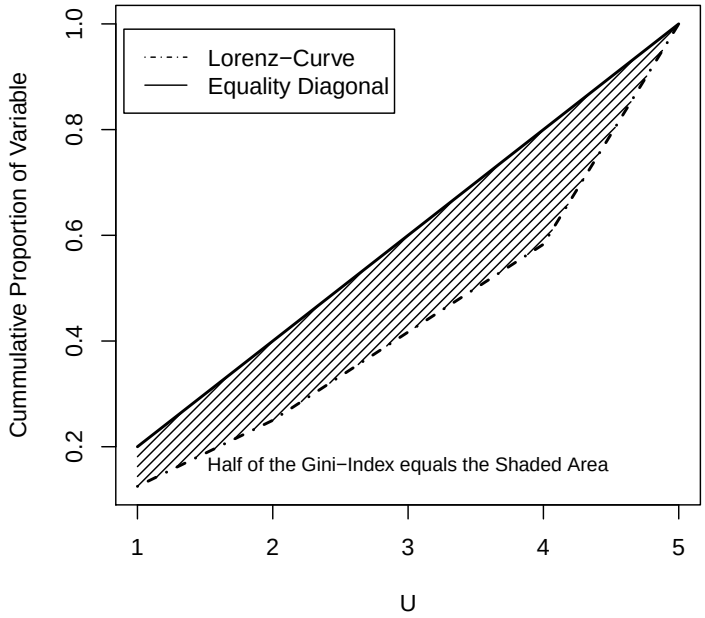

Figure 5.2: Lorenz Curve and Gini-Index (plottet via the function in [Faes, 2007])

The Gini-Index describes now the deviation of the Lorenz-Curve from the curve of the perfect equality which is represented as a diagonal line. The area between this line and the Lorenz-Curve is the half of the Gini-Index. For the **Example** above the Gini-Index is 0.25.
For more information see [Travis, 2008] and [Sachs and Hedderich, 2006].

Definition giniUTR

The first indicator is supposed to compare the upper-to-total-ratio of the data with the upper-to-total-ratio of the uniform distribution for m level cuts and hence the Gini-Index can be used as measure.
Using the UTR in Equation (5.1) and the Definition (5.2) of the Gini-Index yields the following definition of the first indicator

$$giniUTR = gini(UTR) = \frac{\sum_{i=1}^{n}(2i - n - 1)UTR_i}{n^2\mu}, \tag{5.3}$$

where $UTR_i \in [0,1]$ and $\mu = \frac{1}{n}\sum_{i=1}^{n} UTR_i$.

5.2 Indicator 2: cpUTR

The second indicator is a measure for the comparison of the behaviour of the UTR of the data with the UTR of the normal distribution for several level cuts $\boldsymbol{\alpha} \in \mathbb{R}^m$ along the z-axis. For this purpose the Gini-Index again will be considered for both UTR of both distributions.

The UTR of the bivariate normal distribution with the parameters $\rho = 0$, $\mu_1 = \mu_2 = 0$ and $\sigma_1 = \sigma_2 = \sigma$ can also be calculated analytically, see in Chapter 4, (4.10) and (4.11). Since the contours of the bivariate normal distribution describe in this case circles with center $(0,0)$ and radius $r_i = \sqrt{-2\sigma^2 \ln(2\pi\sigma^2\alpha_i)}$ for several level cuts α_i with $i = 1,\ldots,m$, the UTR_i can be calculated as the ratio of the area of the circle at the level cut α_i in relation to the area of the circle at the first level α_1. Therefore the UTR of the bivariate normal distribution with the parameters $\rho = 0$, $\mu_1 = \mu_2 = 0$ and $\sigma_1 = \sigma_2 = \sigma$ is defined as

$$nUTR_i = \frac{r_i^2}{r_1^2}, \quad i = 1,\ldots,m, \tag{5.4}$$

where $nUTR$ is a m-dimensional vector with $nUTR_i \in [0,1]$.

For this bivariate normal distribution the $nUTR$ can also be defined as a function, see below.

Derivation of the $nUTR$ for the $N(0,0,\sigma^2,\sigma^2,0)$

At first the equidistant level cuts α_i with $i = 1,\ldots,m$, where $a = \alpha_1 < \alpha_2 < \ldots < \alpha_m = b$ are defined as a function k through

$$k(i) = a + (i-1)\frac{b-a}{m-1}$$

$$= a + (i-1)h_\alpha. \tag{5.5}$$

Considering $r_i = \sqrt{-2\sigma^2 \ln(2\pi\sigma^2\alpha_i)}$ and (5.4) yields the following definition of a function for the $nUTR$

$$nUTR_i = \frac{r_i^2}{r_1^2}$$

$$= \frac{-2\sigma^2 \ln(2\pi\sigma^2\alpha_i)}{-2\sigma^2 \ln(2\pi\sigma^2\alpha_1)}$$

$$\overset{by(5.5)}{=} \frac{-2\sigma^2 \ln(2\pi\sigma^2 k(i))}{-2\sigma^2 \ln(2\pi\sigma^2 k(1))}$$

$$= \frac{\ln(2\pi\sigma^2 k(i))}{\ln(2\pi\sigma^2 k(1))}$$

$$:= V(k(i)). \tag{5.6}$$

Now it is possible to calculate the derivative of the function $V(k(i))$ and hence the behaviour of the $nUTR$ can be analyzed. Using the chain rule

$$\frac{\partial V(k(i))}{\partial i} = \frac{\partial V}{\partial k} \cdot \frac{\partial k}{\partial i}$$

the derivative of V is

$$\frac{\partial V(k(i))}{\partial i} = \frac{1}{2\sigma^2\pi \ln(2\pi\sigma^2 k(1)) \cdot k(i)} \cdot \frac{\partial k}{\partial i}, \tag{5.7}$$

where

$$\frac{\partial k}{\partial i} = h_\alpha = \frac{b-a}{m-1} \quad > 0, \tag{5.8}$$

and thus

$$\frac{\partial V(k(i))}{\partial i} = \frac{h_\alpha}{-r_1^2\pi}\frac{1}{k(i)}$$

$$= -\frac{h_\alpha}{r_1^2\pi(a+(i-1)h_\alpha)} \quad < 0 \quad \forall i \in \mathbb{N}.$$

If $i \in \mathbb{N}$ increases by one unit, then k will increase with the factor h_α and hence V will decrease with $-\frac{h_\alpha}{r_1^2\pi(a+(i-1)h_\alpha)}$ towards 0.

The following definition gives an indicator for the comparison of the UTRs between the data and the bivariate normal distribution.

Definition cpUTR

The absolute value of the difference of both Gini-Indices is calculated as

$$cpUTR = |gini(UTR \text{ of data}) - gini(UTR \text{ of normal distr.})|. \tag{5.9}$$

Figure 5.3 demonstrates an example for the second indicator $cpUTR$, where the shaded area corresponds to the $cpUTR$.

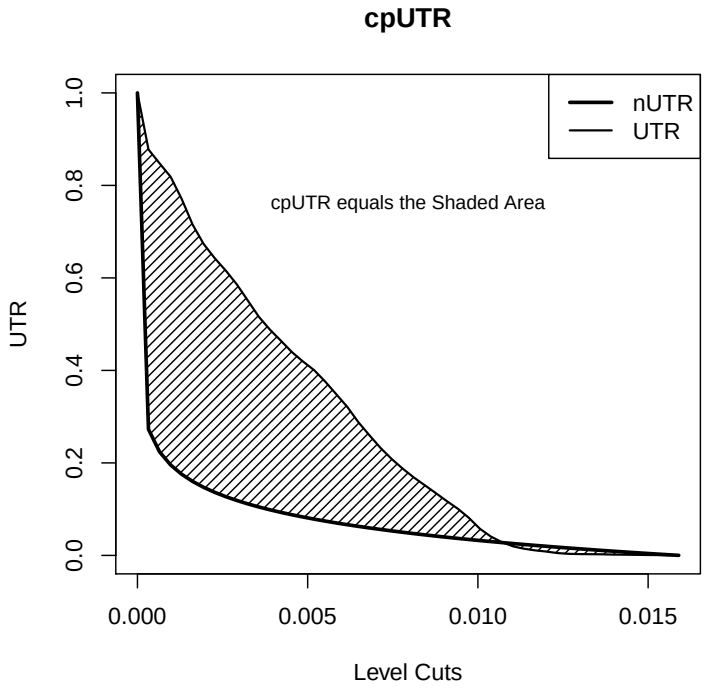

Figure 5.3: Plots of the UTR of the bivariate normal distribution and the UTR of a density estimation.

5.3 Indicator 3: NGroups

The following indicator is supposed to give a view into the behaviour of the groupings within the density estimation for each level cut α_i, with $i = 1, \ldots, m$, within a predetermined evaluation area (eval.area).

eval.area

The evaluation area is the largest coherent area containing the majority of the density. For the determination of the evaluation area, the largest contour line of the density estimation at the level of a given baseline (like in Chapter 4, Equation (4.9)) is selected. The purpose is to determine the number of groupings of the density values within this eval.area for several level cuts, meaning the number of the yellow areas in the right plot in Figure 5.1.

Definition NGroups

The indicator *NGroups* is defined as the total number of the groupings within the eval.area at each level cut α_i:

$$NGroups_i = \sum \mathbb{I}(G_j \in eval.area), \tag{5.10}$$

with

$$\mathbb{I}(G_j \in eval.area) = \begin{cases} 1, & \text{for } G_j \in eval.area, \\ 0, & \text{else }, \end{cases} \tag{5.11}$$

where G_j denotes the j-th grouping, with $j = 1, \ldots, k$.

The calculation of *NGroups* for several level cuts $\boldsymbol{\alpha} \in \mathbb{R}^m$ is of further interest and will also be needed for the calculation of the last indicator $modCHI$ (5.17). Thus *NGroups* is used to be a vector of length m, see also Chapter 6.

5.4 Indicator 4: modCHI

The second part of the analysis of the distribution of the measurements deals with the spatial distribution of the groupings among them.
In the previous subsection the number of the groupings has been defined. Now the separation of these groupings of the density is of interest. Therefore the aim now is to find a measure to set the density groupings in relation to the distances between the groupings. An appropriate method is a modification of the Calinski Harabasz Index, which is explained in detail below.

Calinski Harabasz Index

The Calinski Harabasz Index (CHI) is used here to be a measure for the (dis)similarity between groupings over the (dis)similarity within groupings, see [Calinski and Harabasz, 1974], [Maulik and Bandyopadhyay, 2002] and [Schlittgen, 2009].

Thus the sum of squared errors between the j-th grouping and the remaining $j-1$ groupings is calculated and compared to the within sum of squared errors for the j groupings.

$$BSS(k) = \sum_{j=1}^{k} n_j(\overline{x}_j - \overline{x})(\overline{x}_j - \overline{x})^T \tag{5.12}$$

is the between sum of squares for k groupings, where $\overline{x}_j = \frac{1}{n_j}\sum_{i=1}^{n_j} x_i$ is the center of G_j and n_j is the number of data points within grouping G_j and $\overline{x}$ is the overall mean. The within sum of squares is as follows

$$WSS(k) = \sum_{j=1}^{k} \sum_{x_i \in G_j} (x_i - \overline{x}_j)(x_i - \overline{x}_j)^T. \tag{5.13}$$

Hence, the Calinski Harabasz Index is defined as

$$CHI = \frac{BSS(k)}{WSS(k)} \frac{n-k}{k-1}, \tag{5.14}$$

where n is the number of data points and k is the number of groupings.

A larger value for CHI as dissimilarity measure indicates a better separation, because this means that the BSS has a high value and the WSS a lower one and hence the difference between the groupings is large.

Definition modCHI

A slight modification of the Calinski Harabasz Index yields the definition of the last indicator.
Instead of the BSS and WSS of the k groupings, the sum of the absolute distances between the centers of the groupings (BSA) and the sum of the absolute distances of the density points from the center of the grouping within the groupings (WSA) is considered:

$$BSA = \sum_{l>j=1}^{k} |\overline{x}_l - \overline{x}_j| \tag{5.15}$$

and

$$WSA = \sum_{j=1}^{k} \sum_{i=1}^{n_j} |x_i - \overline{x}_j|, \tag{5.16}$$

where n_j is the number of density points within grouping j and $\overline{x}_j = \frac{1}{n_j} \sum_{i=1}^{n_j} x_i$ is the center of the j-th grouping.

Analogously to the CHI, the indicator $modCHI$ is obtained by setting BSA and WSA in relation.

$$modCHI = \frac{BSA}{WSA} \frac{N-k}{k-1}, \tag{5.17}$$

where N is the number of the density points within the considered area and k is the number of groupings ($NGroups$) within this area.

Since the $modCHI$ is a dissimilarity measure, the larger the values for the $modCHI$ the better the separation of the groupings within the considered area.

The indicator $modCHI$ is also used to be a vector with a length of m, since it is calculated for several level cuts $\boldsymbol{\alpha} \in \mathbb{R}^m$ within the predetermined eval.area, see Chapter 6.

Chapter 6

Evaluation

In this chapter the results of the computer-based statistical analysis of two human brain tumor tissue sections with **R** will be presented. Since the data size of the original digitalized slides are in average about 100.000x100.000 pixels and hence too big for the analysis (see Chapter 3.2), the sample-files will be considered and analyzed partly. Therefore the slides are splitted into several sectors at first. Both samples have been previously aparted into several sectors of size 5000x5000 pixels, like explained in Chapter 3.2.

The data for each sector, which has to be investigated, consists of three columns, where the first and second column contain the x- and y-coordinates of the pixels of the Ki67 labelled cell nuclei. The last column represents the categories of cell nuclei.

The next step is to perform a two-dimensional kernel density estimation with a bivariate Gaussian kernel (Chapter 4.2) for each sector by using the predefined function **kde2d()** within the **R** - package **MASS** [R, 2011] . To find an acceptable bandwidth, a function named `Bandwidth()` (Appendix A.2) has been written. This function computes the wanted bandwidth h for all sectors of both samples by evaluating the optimal bandwidth for each sector using the method "better rule of thumb" in Chapter 4.2 Equation (4.31), and then taking the mean of the calculated optimal bandwidths.
The two-dimensional kernel density estimation is evaluated on a square grid, where the number of grid points in each direction has been chosen as 100. The obtained density estimations of each sector have been saved in a list to simplify further analysis.

In the following only the values of the density estimations of the sectors, greater than a baseline, a defined density level, will be considered for the analysis since the values beneath the baseline are too close to zero and hence negligible.

Therefore an **R**-function named `baseline()` (Appendix A.5) has been written. This function evaluates the levels of the contour lines of the density estimations for all sectors of both samples and takes the minimal level of all levels as baseline *bl* - also called as minlevel. To compute those contour lines the predefined **R**-function in the package **grDevices** called **contourLines()**, [R, 2011], has been used. This function calculates the coordinates of the contours for one or more levels for a set of data, consisting of the x- and y- coordinates of the grid lines and the corresponding measurements at the grid points.

For the further analysis and the application of the indicators defined in Chapter 5, an unitary vector of level cuts $\boldsymbol{\alpha} \in \mathbb{R}^m$ along the z-axis of the two-dimensional density estimation for each sector per sample has to be determined. Therefore an equidistant vector of $m = 50$ level cuts is defined, starting at the baseline up to a maximum.

6.1 Evaluation of Sample 1

The statistical analysis takes place in two steps. During the first one the aim is to get any information about the behaviour of the distribution of the data. This purpose is satisfied by the indicators $giniUTR$ and $cpUTR$, previously defined in Chapter 5. The second part of the analysis deals with the spatial distribution of groupings among the values of the density estimations. The indicators $NGroups$ and $modCHI$ deliver the corresponding measures.

The first Sample of a human brain tumor tissue section, that is going to be analyzed, has been splitted into 20 equal sectors of size 5000x5000 pixels during the preprocessing (see Chapter 3.2) and delivered 14 sectors which are relevant for further analysis.

Figure 6.1 shows the original scanned and digitalized tumor tissue slide, whereby the grid, which splits the sample, is also added. The brown points represent the cell nuclei of the proliferating cells.

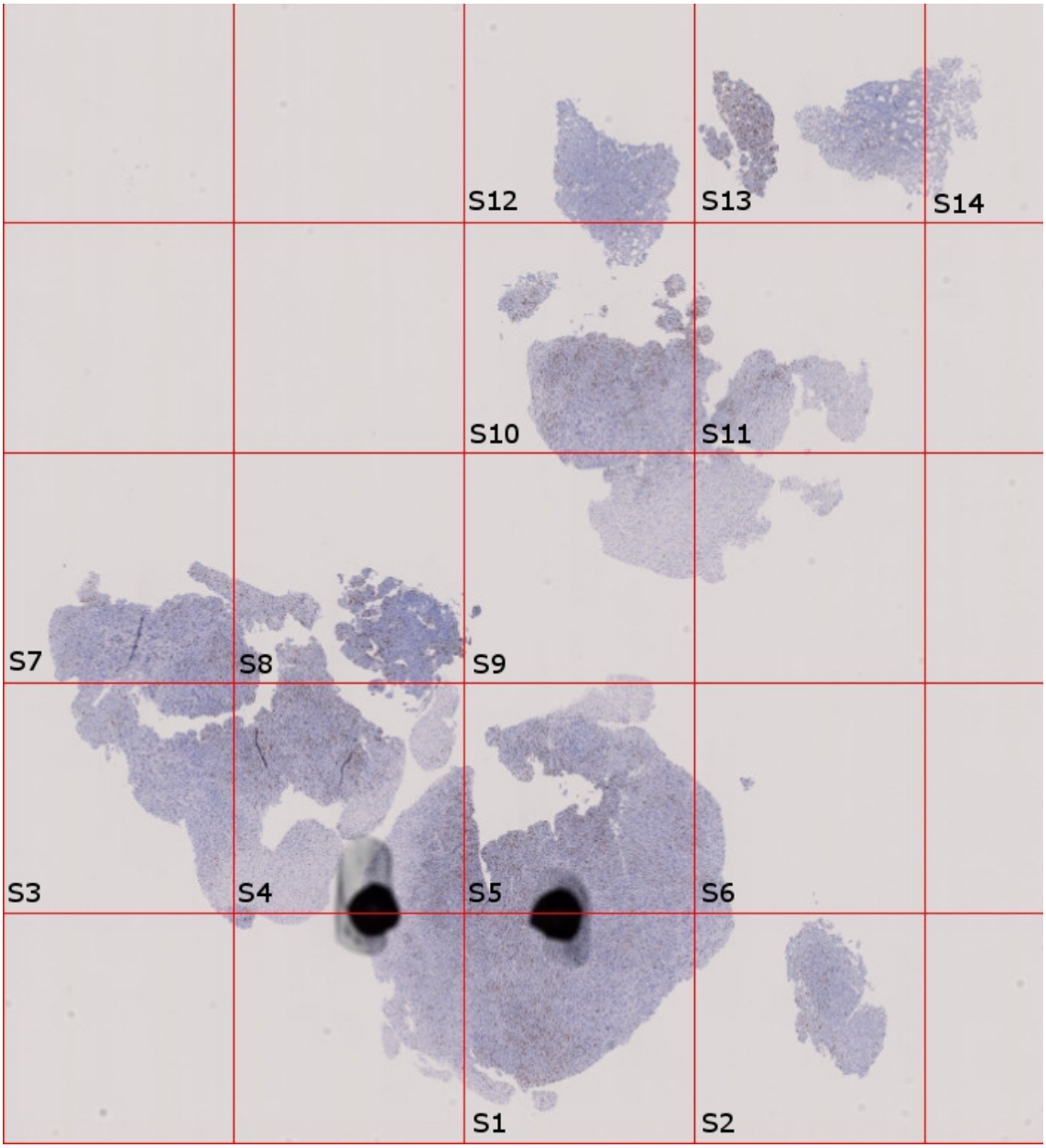

Figure 6.1: Map of Sample 1 - NDPI_1250

Starting with sector S_1 all sectors will be analyzed in the following.

Figure 6.2 shows the plot of Sector S_1 in $\mathbf{R}$, where the black points represent the Ki67 labelled cell nuclei.

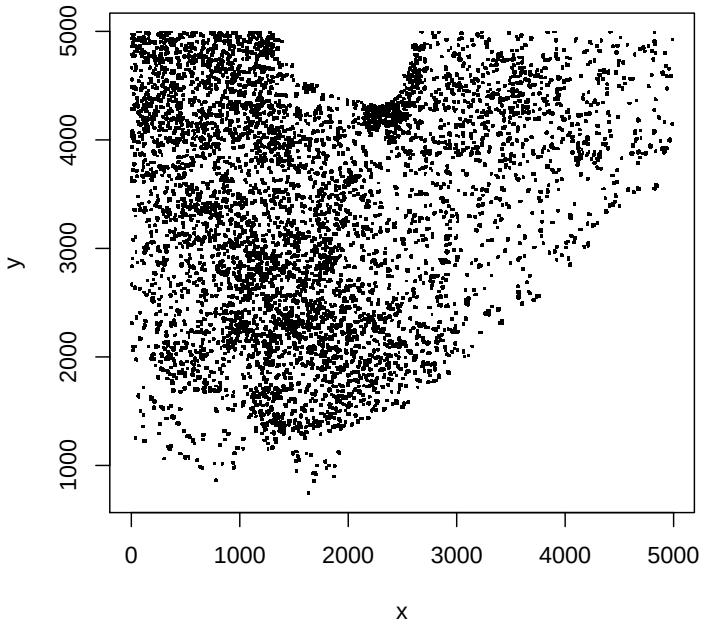

Figure 6.2: Marked Cell nuclei of Sector S_1

In Figure 6.3 a two dimensional kernel density estimation with a Gaussian kernel and the optimal bandwidth $h = (541.12, 544.90)^T$, calculated like in Equation (4.31) with the function `Bandwidth()` (Appendix A.2), is presented.

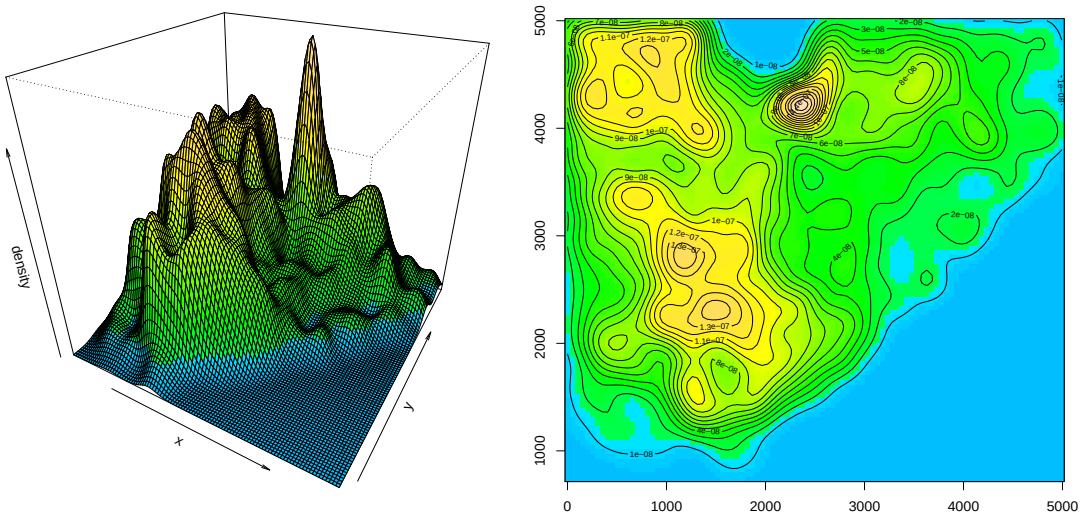

Figure 6.3: Kernel Density Estimation of Sector S_1

6.1.1 Evaluation Indicator giniUTR

The inequality in the behaviour of each density estimation of each sector per 50 level cuts is analyzed, i.e. the deviation from the perfect uniform distribution. For the calculation of the density of the theoretical bivariate uniform distribution two vectors of size 100, with components in the interval $[0, 100]$, have been considered (see `gvt2d()`, Appendix A.6).

Definition Level Cuts

A vector of 50 equidistant level cuts $\alpha_j \in [bl, ul]$ is defined, where bl is the baseline calculated through the function `baseline()` (Appendix A.5) and the upper limit ul is the maximum value of the density estimations of all regarded sectors and is defined as

$$ul = \max_{i=1,\dots,n} (d_{S_i}), \tag{6.1}$$

where d stands for the density estimations of the sectors with n is the total number of the considered sectors of both samples.

Thus the vector of level cuts $\boldsymbol{\alpha}$ has components within the interval $[1.00e-08, 1.92e-06]$.

UTR of Sample 1

In the following the UTR for all 14 Sectors of Sample 1 is calculated, like in Chapter 5 in Equation (5.1), by using the function `UTR()` within the function `Inequ()` in **R**, see Appendix A.8 and Appendix A.9.

The figures below show the results for the UTR of all sectors of Sample 1:

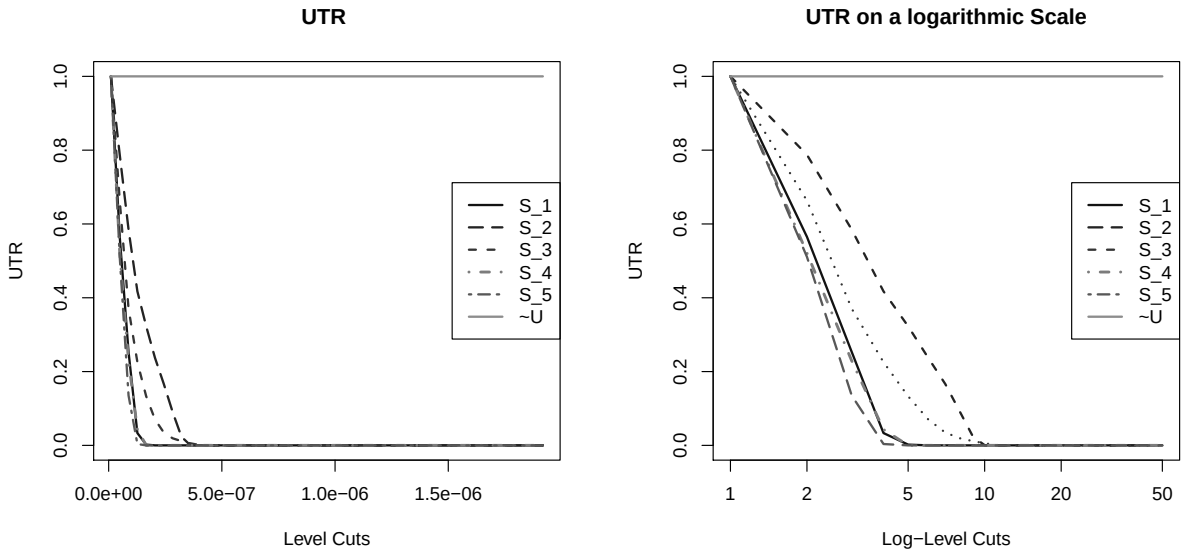

Figure 6.4: UTR for Sector S_1 to S_5

Figure 6.4 (left) presents the UTR for the Sectors 1 to 5, while the figure in the right shows the UTR for those sectors on a logarithmic scale. The solid grey line indicates the UTR for the bivariate uniform distribution $U(0, 0, 100, 100)$.
The UTRs for the Sectors S_1, S_4 and S_5 seem to behave similar, nearly identical with a fast decrease to 0, whereas the Sectors S_2 and S_3 show a significant deviation.

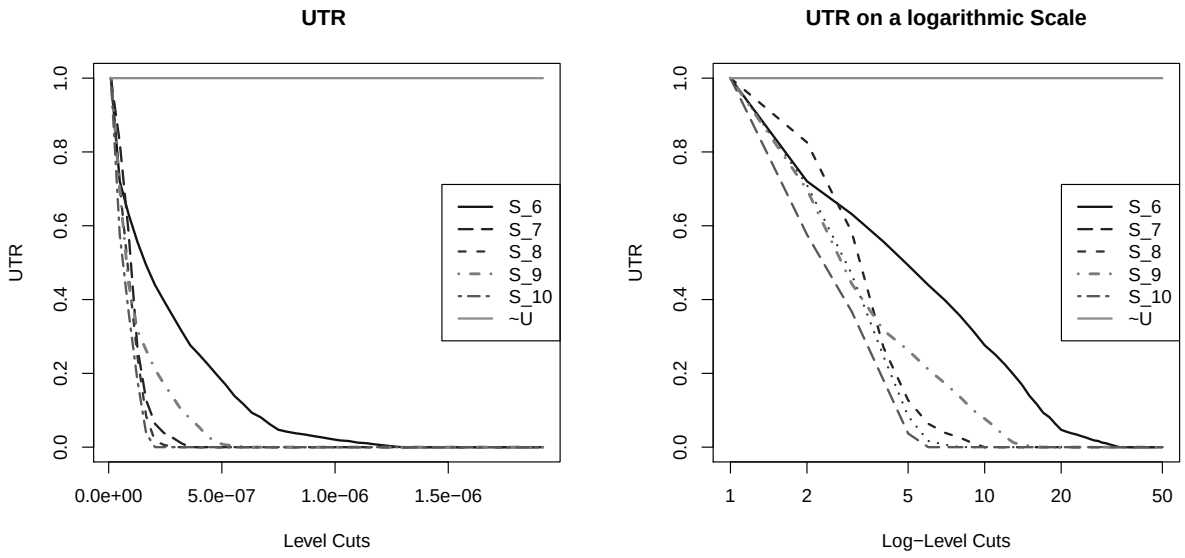

Figure 6.5: UTR for Sector S_6 to S_{10}

From Figure 6.5 it is obvious that the UTR of Sector S_6 falls slower than in other sectors.

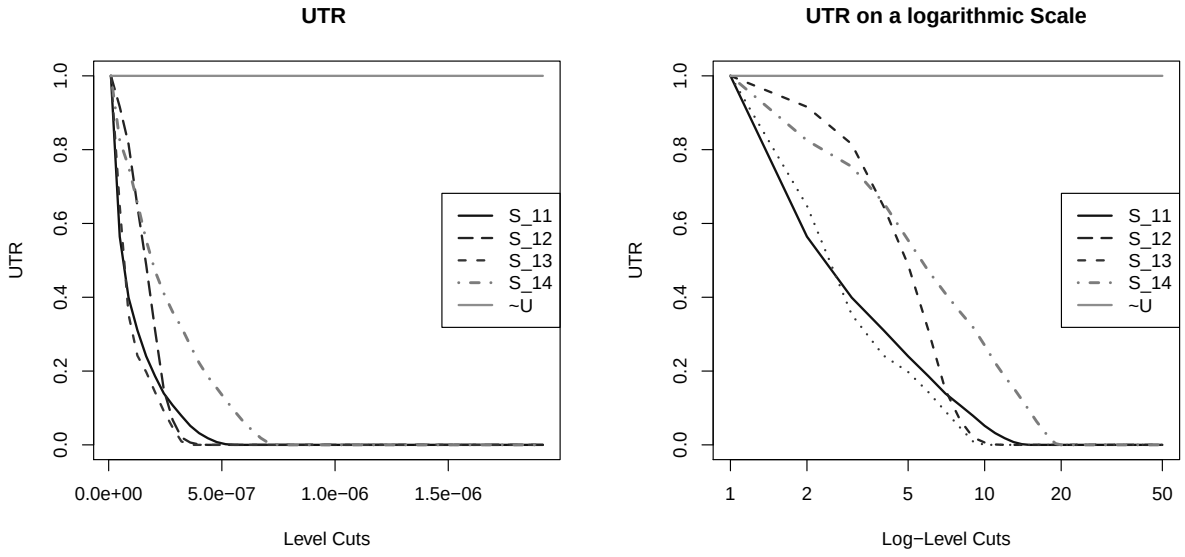

Figure 6.6: UTR for Sector S_{11} to S_{14}

The Sector S_{14} has also a slower decrease than the other sectors. The comparison of

the UTR of all sectors of Sample 1 shows that the Sectors S_2, S_6, S_{12} and S_{14} have a significant deviation in the behaviour than the remaining ones.

giniUTR of Sample 1

The indicator $giniUTR$ calculates the inequality within a vector, in other words the deviation of this vector from the perfect uniform distribution.
This is ensured through the function `Inequ()`, see Appendix A.9, and is calculated like explained in Chapter 5 in Equation (5.3).

If the $giniUTR$ is 0, there is no deviation from the uniform distribution, meaning there is a perfect equality. $giniUTR = 1$ means that there is high concentration to one value.

The following table gives the percental inequality in the UTR of all sectors of Sample 1:

Sector	1	2	3	4	5	6	7
giniUTR in %	95.45	89.73	92.54	95.52	96.08	73.35	92.39
Sector	8	9	10	11	12	13	14
giniUTR in %	93.50	87.47	94.29	88.43	89.32	91.35	79.72

Table 6.1: giniUTR of all sectors of Sample 1

Overall, the obtained results show that there is a concentration to one value in all sectors of Sample 1. Sector S_5 shows the largest inequality with 96.08% and the smallest inequality is in Sector S_6 with 73.35%.

6.1.2 Evaluation Indicator cpUTR

The second indicator gives a measure for the deviation from the behaviour of the bivariate normal distribution. Therefore the UTRs of all sectors again are considered and compared

with the UTR of the normal distribution.

For this purpose two vectors of size 100 within the interval $[-10, 10]$ have been defined. Following parameters as mean and standard deviation for the bivariate normal distribution have been selected: $\mu_1 = \mu_1 = 0$ and $\sigma_1^2 = \sigma_2^2 = 10$, where the correlation is $\rho = 0$. The calculation of the density in **R** is done through the function `nvt2d()` (Appendix A.7).

The next step is to scale the density estimations of the sectors and redefine the level cuts $\boldsymbol{\alpha}$ respectively. This is necessary since the aim here is the comparison with the behaviour of the normal distribution.
The level cuts are now within the interval $[bl, max(N(0, 0, 10, 10, 0))]$ and hence $\alpha_j \in [1e-08, 0.0159]$, with $j = 1, \ldots, 50$. Again 50 equidistant level cuts are considered.

The scaling of the density estimations of the sectors is done in **R** during the calculation of the UTR within the function `Inequ()`, see Appendix A.9.

The UTR for the bivariate normal distribution is calculated like in Chapter 5 in Equation (5.6) via the **R**-function `nUTR()`, see Appendix A.10.

The following figures show the scaled UTRs of all sectors of Sample 1 including the UTR for the bivariate normal distribution.

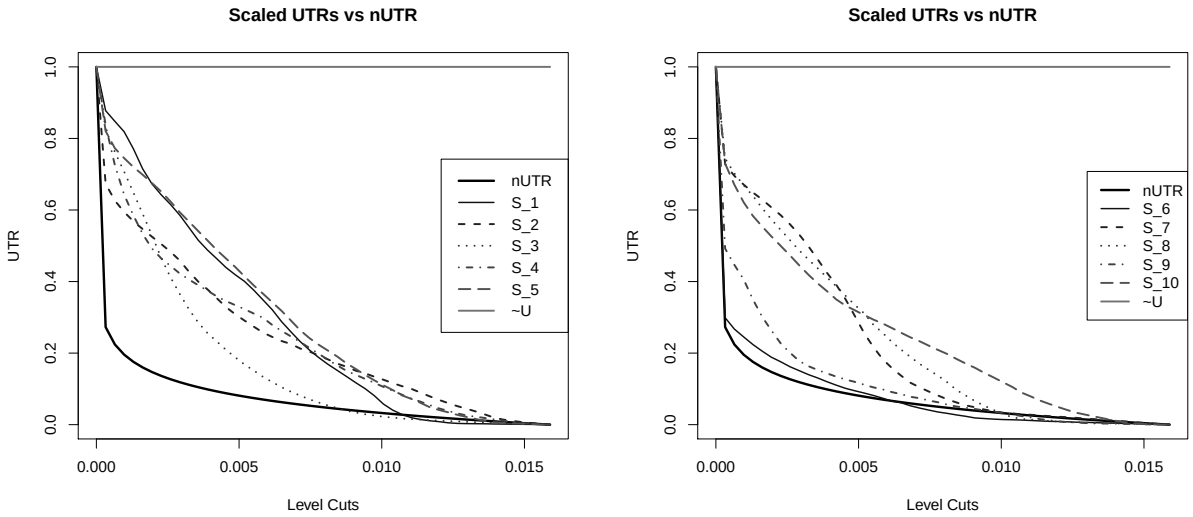

Figure 6.7: Scaled UTRs for Sector S_1 to S_{10}

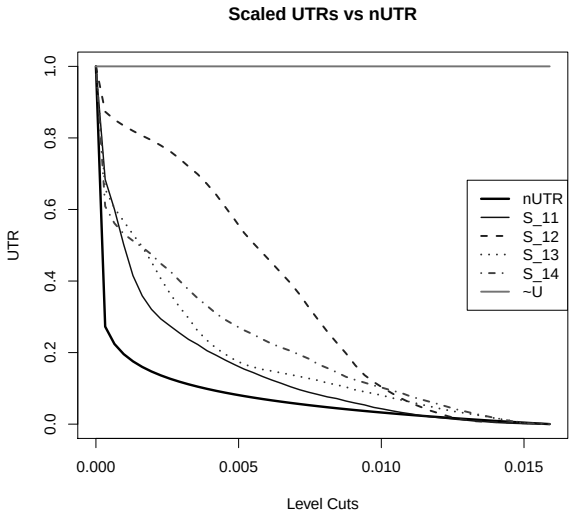

Figure 6.8: Scaled UTRs for Sector S_{11} to S_{14} vs. $nUTR$

From Figure 6.8 it is visible that the scaled UTR of Sector S_{12} of Sample 1 shows the most deviant behaviour of the $nUTR$ and of the other sectors, whereas Sector S_6 has the most similar behaviour of the UTR to the $nUTR$. Sectors S_9 and S_{11} show also an UTR near to the UTR of the normal distribution.

In the following Table 6.2 the results for the comparison of the scaled UTRs of the sectors

with the $nUTR$ of the bivariate normal distribution are presented, where the computations are done in **R** like explained in Chapter 5 in Equation (5.9), by using the function `CompNdist()`, see Appendix A.11.

Sector	1	2	3	4	5	6	7
$cpUTR$ in %	1.39	10.20	8.21	7.44	6.71	7.36	2.74
Sector	8	9	10	11	12	13	14
$cpUTR$ in %	1.19	5.45	8.47	2.81	6.94	3.63	8.56

Table 6.2: Results for $cpUTR$ of Sample 1

The obtained results show that the maximum absolute difference in behaviour of the UTRs is between the Sector S_2 and the normal distribution and is 10.2%. The minimum difference is between S_8 and normal distribution and S_1 and normal distribution. In average there is only a slight difference in behaviour of Sample 1 and the bivariate normal distribution.

6.1.3 Evaluation Indicator NGroups

The following two indicators deal with the spatial distribution of possible groupings within the density estimations of all sectors. Therefore 50 equidistant level cuts $\alpha_j \in [bl, ul]$, with $j = 1, \ldots, 50$ are considered again and thus $\alpha_j \in [1.00e-08, 1.92e-06]$.

The indicator *NGroups* calculates the number of groupings per level within a defined coherent evaluation area, see Chapter 5 Eval.area 5.3) and Equation (5.10).
Hence the next step is to define such an area. This is enabled in **R** through the function `Areanew()`, see Appendix A.13. This function has as input the largest contour line of the density estimation at the level of baseline and the density estimation and determines, considering several conditions, the coordinates of the requested evaluation area. This step is necessary since the predefined function `contourLines()` of **R**, which calculates the coordinates of the contours of a data set, was inadequate for the data to investigate. Thus an extension of the area determined via `contourLines()` was required and complied through `Areanew()`, see Appendix A.13.

Next the grouping within a density estimation has to be determined. Therefore another function must be used, called `grouping()` (Appendix A.15). In this function the groupings within the density estimation per level cut is determined and then the values of the density estimation are assigned to the corresponding groups.

Thereby it is possible now to compute the number of groupings per level for each Sector S_i of Sample 1. The function `NGroups()` (Appendix A.16) delivers the requested result.

The following figures show the obtained results for the indicator *NGroups* for all sectors of Sample 1.

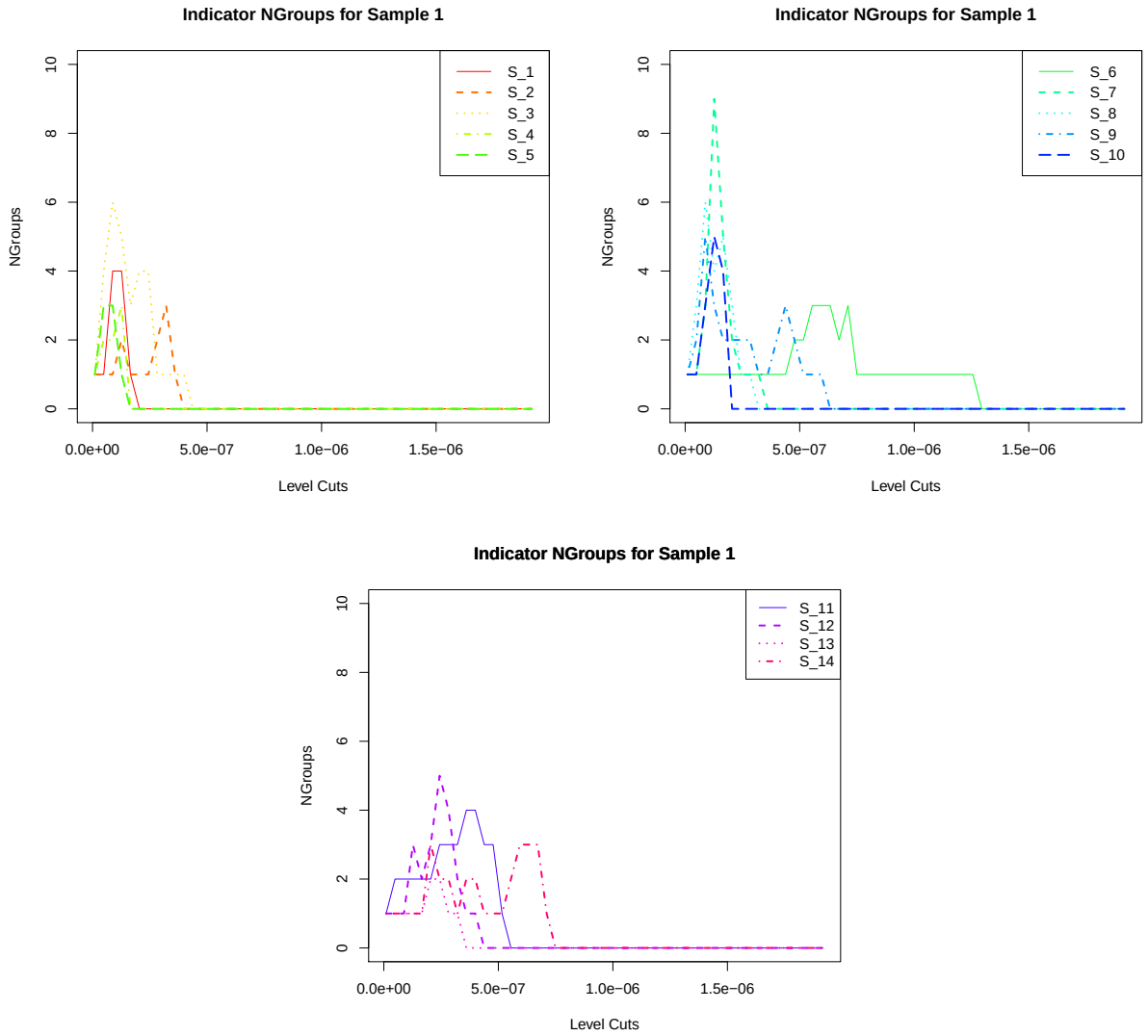

Figure 6.9: Indicator *NGroups* for Sector S_1 to S_{14}

From Figure 6.9 it is apparent, that all sectors start with one grouping at the level of baseline and all end with one grouping at their last level, at which the eval.area is not empty. The largest number of grouping is in Sector S_7 at the 4-th level cut and has the value 9. Sector S_6 is the Sector with the highest density peak, followed by Sector S_{14}.

6.1.4 Evaluation Indicator modCHI

After the computing of the number of groupings per level in the previous subsection, now the separation of these groupings is of further interest. The groupings within the density estimations of the sectors are already determined above, with the function `grouping()` for the calculation of indicator *NGroups*.
The following figures represent the groupings of the density estimation for each sector at the level where the sectors have a maximum of groupings.

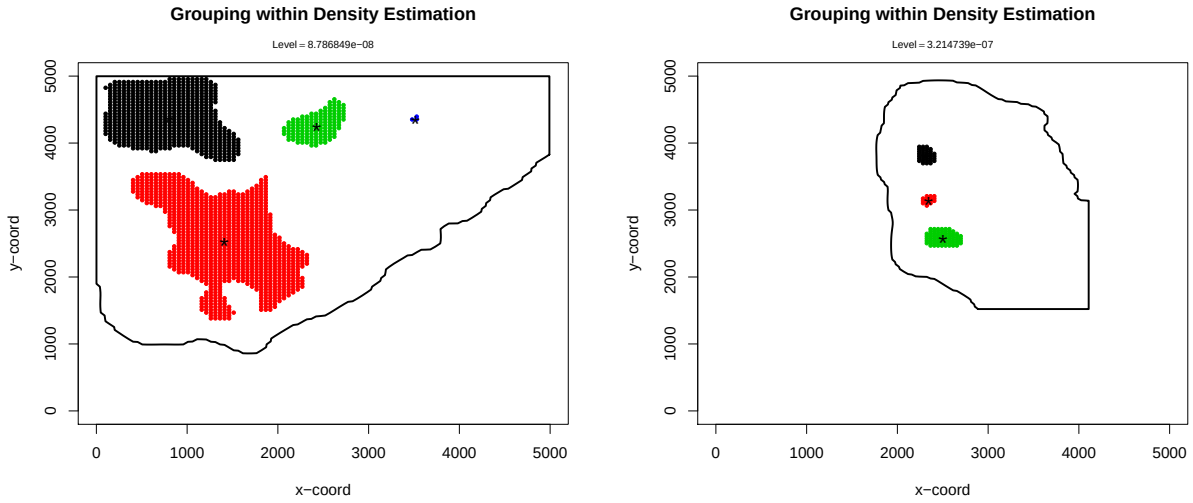

Figure 6.10: Grouping within Sectors S_1 - S_2, where the centers of the groupings are marked with $*$.

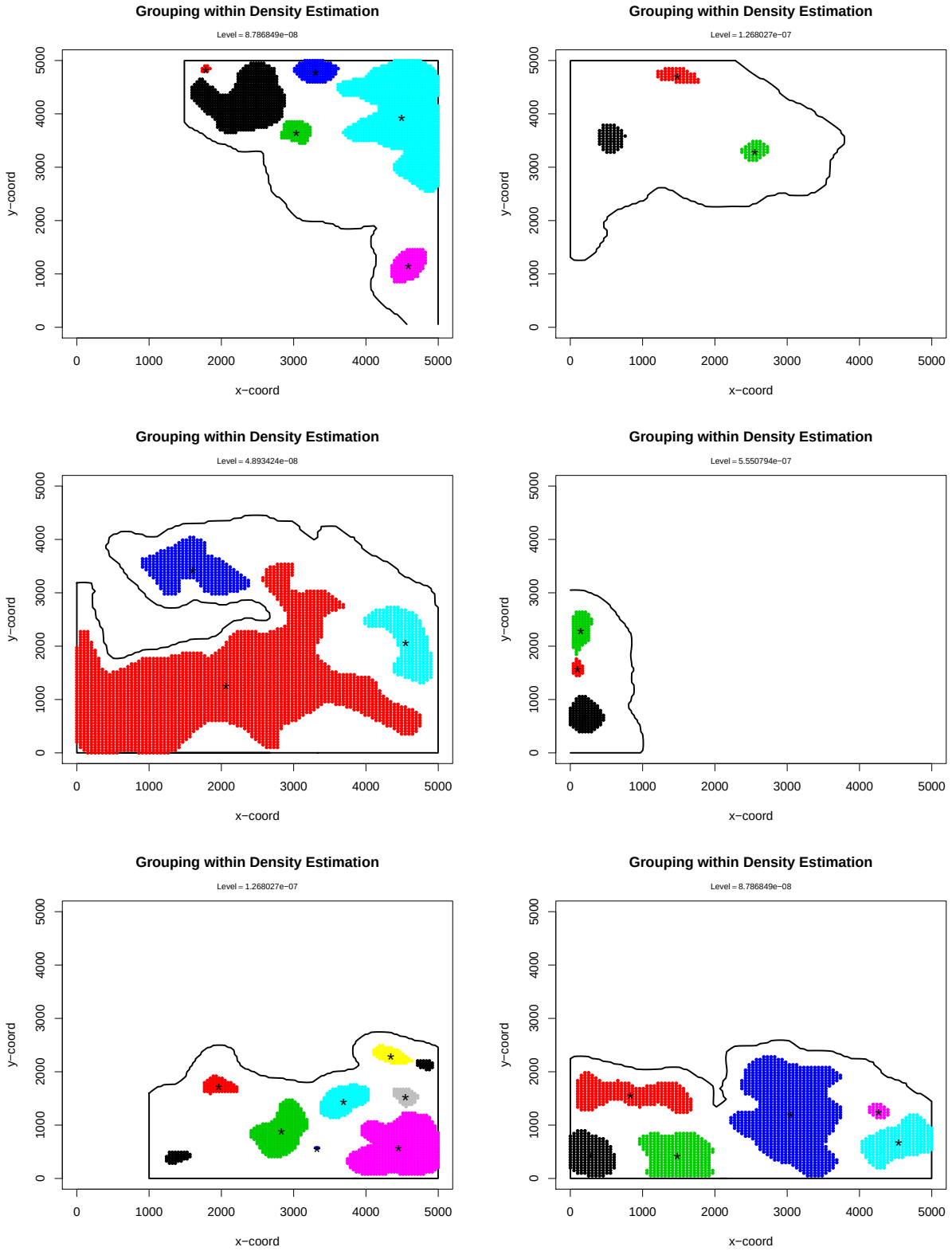

Figure 6.11: Grouping within Sectors S_3 - S_8, where the centers of the groupings are marked with $*$.

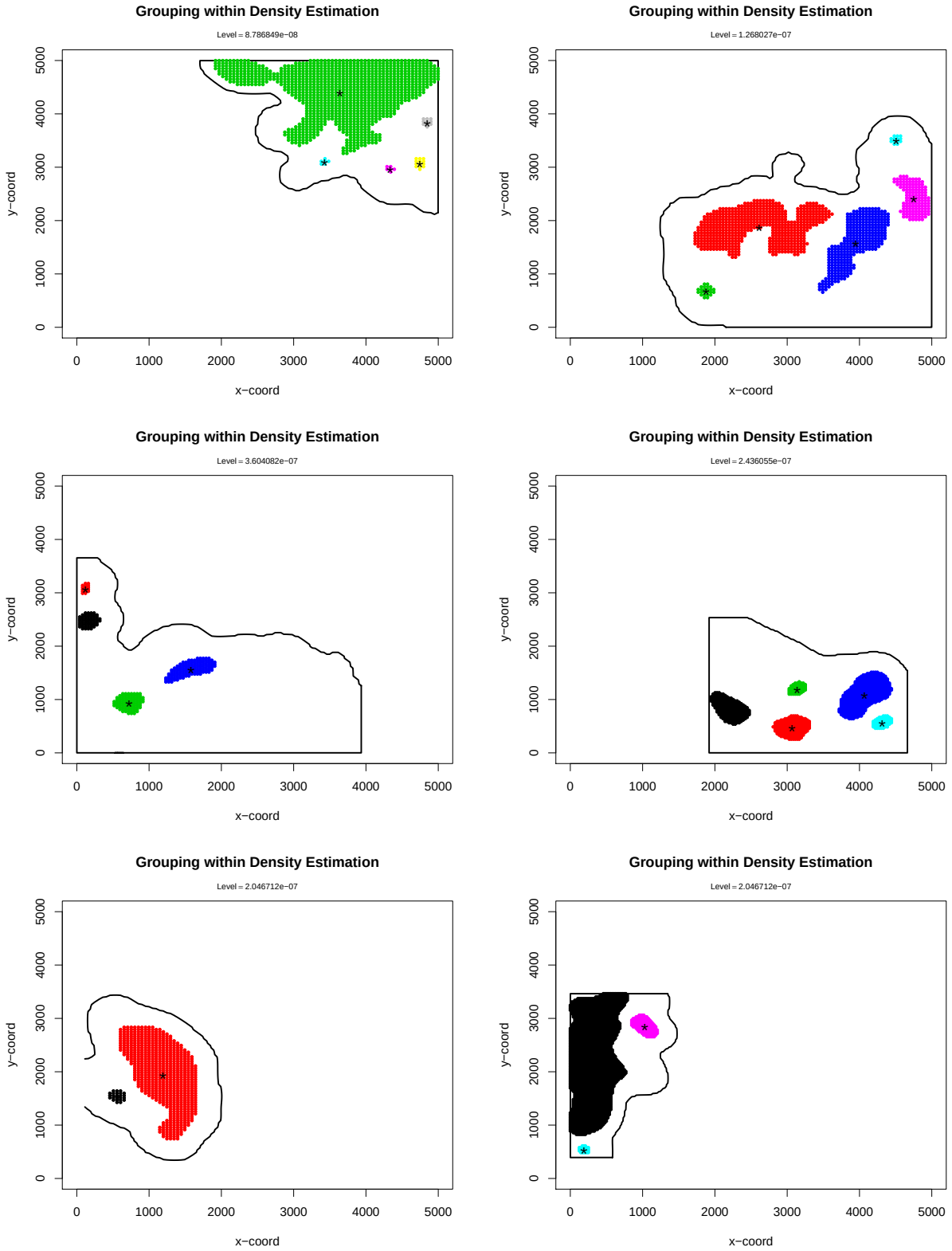

Figure 6.12: Grouping within Sectors S_9 - S_{14}, where the centers of the groupings are marked with $*$.

The center of the k-th grouping is computed through the arithmetic mean via $\overline{x}_k = \frac{1}{n_k} \sum_{i=1}^{n_k} x_i$, where n_k is the number of the density points within the grouping k.

The last indicator $modCHI$ is calculated, like defined in Chapter 5 in Equation (5.17), in **R** via the function `separation()` (Appendix A.17). This function has as input the matrix containing the coordinates of the density estimation with the corresponding groupings at a level α_j and computes as indicator the modified Calinski Harabasz Index by calculating the sum of absolute distances between the centers of the groupings (BSA) and the sum of the absolute distances of the density points from the center of the grouping within the groupings (WSA) with respectively weighted and set in relation.

This function is applied to all sectors considering all 50 levels. The larger the value for the $modCHI$ the better the separation of the groups within the eval.area. Small values would mean that there is a concentration within the evaluation area.

The obtained results for Sample 1 are shown in Figure 6.13:

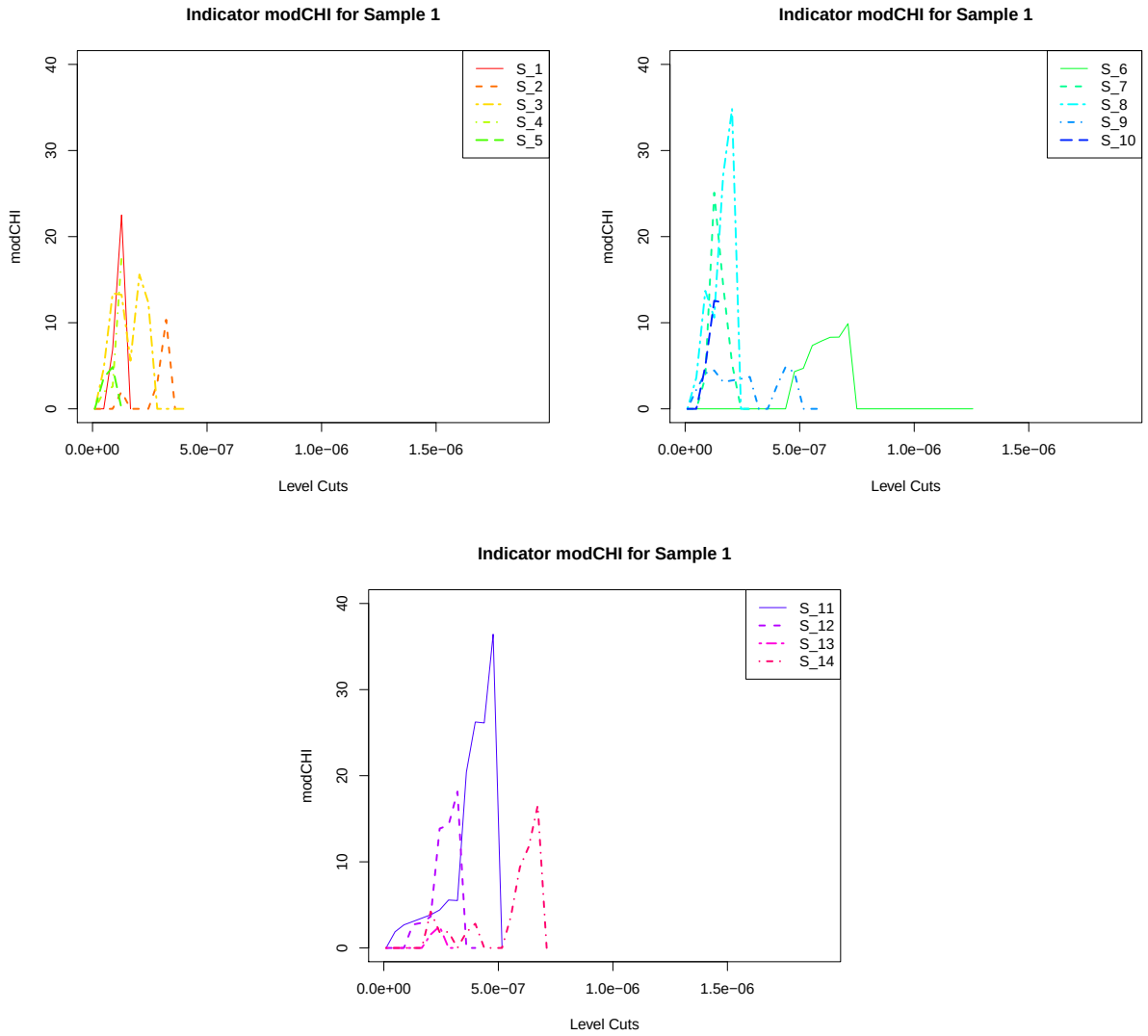

Figure 6.13: Separation of groupings within the eval.area of S_1-S_{14}

The maximum value for $modCHI$ is obtained for Sector S_{11} at the 13-th level being 36.43. $modCHI = 0$ means that there is only one grouping within the eval.area at this level, whereas NA is obtained if there is not any grouping within the eval.area and thus the curves in Figure 6.13 stop at the last level, which is not empty.

6.2 Evaluation of Sample 2

Analogously to the **Evaluation of Sample 1**, Section 6.1, the same indicators are calculated for a second Sample of a digitalized brain tumor tissue section. The results are demonstrated as follows.

After the preprocessing, following image (Figure 6.14) is obtained for Sample 2:

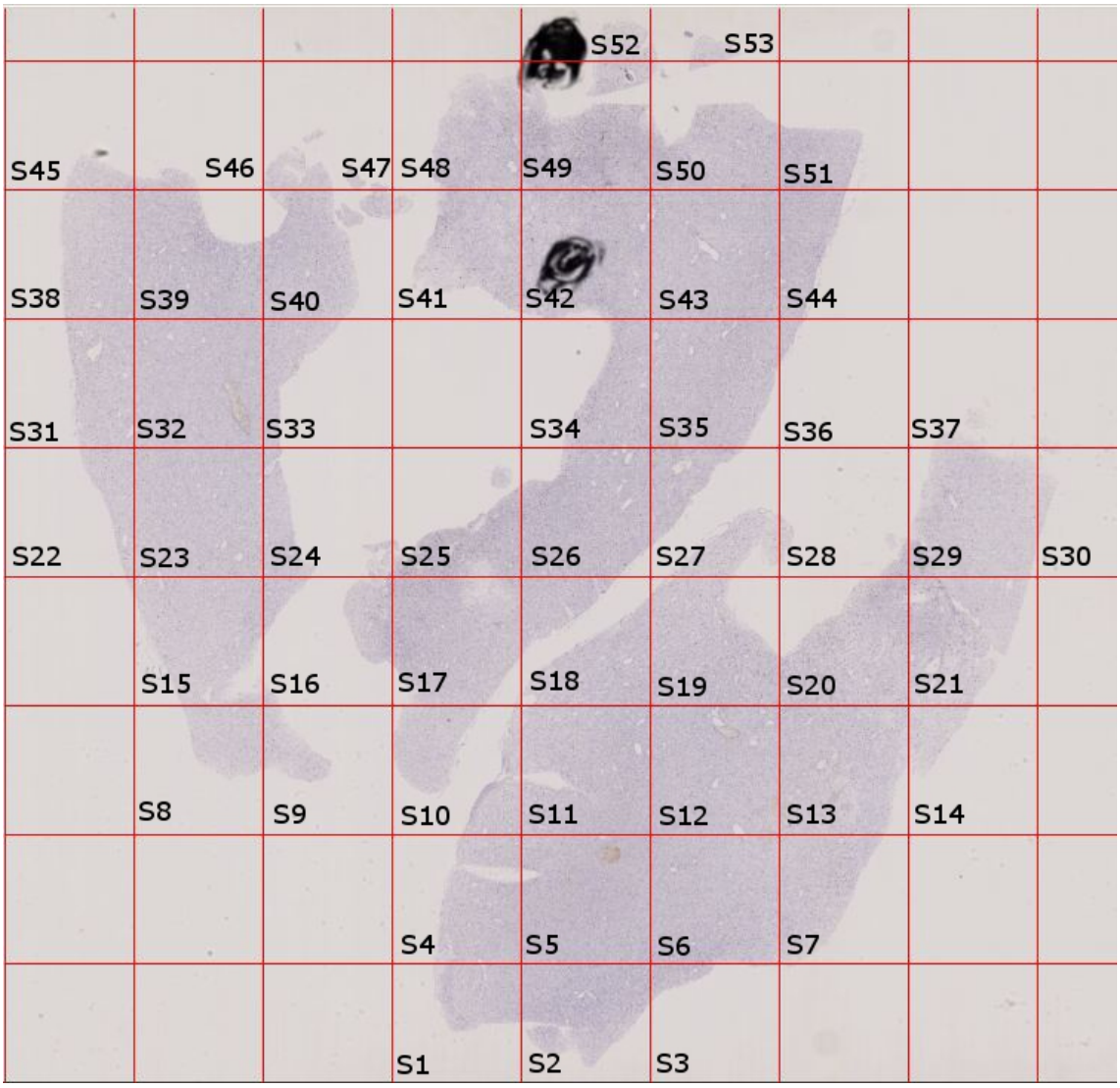

Figure 6.14: Map of Sample 2 - NDPI_1230

Sample 2 is larger than Sample 1 and has therefore more sectors to be analyzed. Sample 2 is aparted into 81 Sectors of size of 5000x5000 pixels, where 53 sectors are selected for further analysis, see Figure 6.14.

Again a two-dimensional kernel density estimation, with a Gaussian kernel and the same bandwidth $h = (541.12, 544.90)^T$ like in Sample 1, is performed for each Sector S_i, $i = 1, \ldots, 53$.

6.2.1 Evaluation Indicator giniUTR

50 equidistant level cuts are considered and the vector of level cuts $\boldsymbol{\alpha}$ has again components in the interval $[1.00e-08, 1.92e-06]$. The inequality in the behaviour of each density estimation of each sector is of interest and thus the UTR for all sectors is compared with the previously computed UTR of the bivariate uniform distribution like in Subsection 6.1 in 6.1.1.

The comparison of the UTRs for the Sectors S_i with $i = 1, \ldots, 53$ of Sample 2 yields that Sector S_{36} has the most deviating behaviour, by showing a slower decrease than the other ones. The Sectors S_8, S_9, S_{14}, S_{16},S_{28}, S_{30}, S_{37}, S_{45}, S_{46} and S_{47} show also a slower decrease. The remaining sectors behave similar and have a fast falling to zero.

The following Figure 6.15 presents the obtained results for the comparison with the bivariate uniform distribution by considering the indicator $giniUTR$.

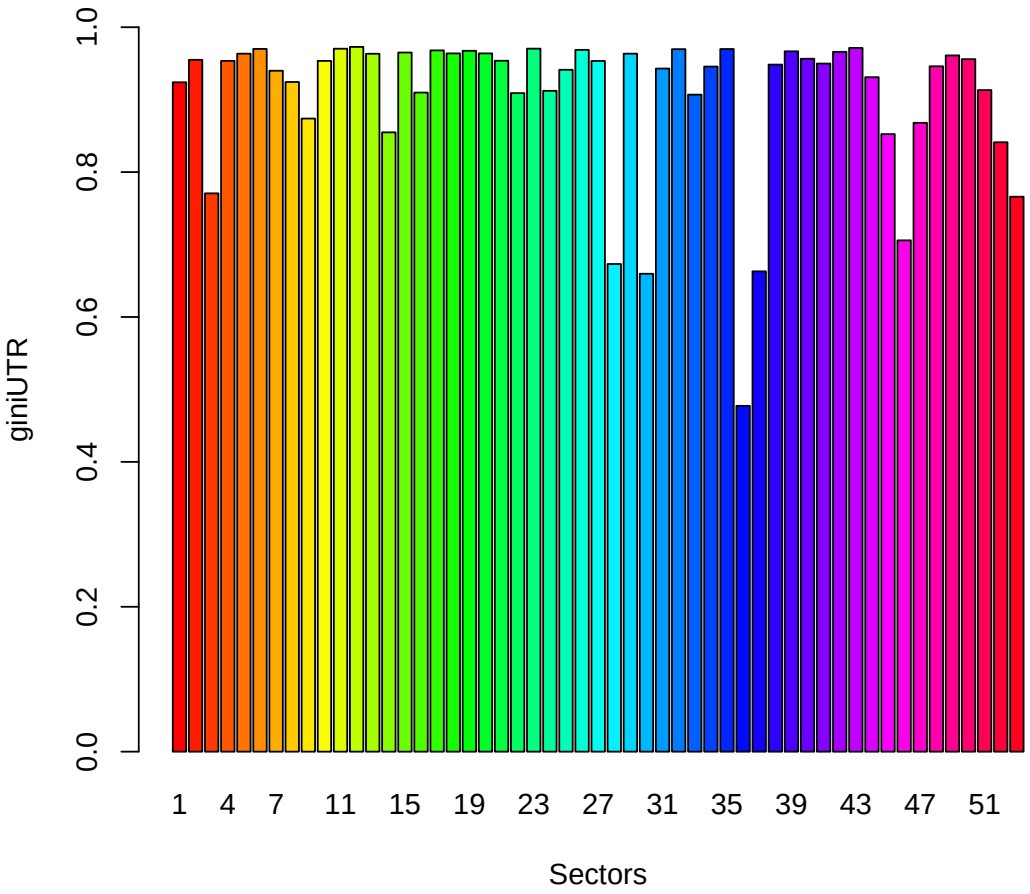

Figure 6.15: Deviation of Sample 2 from the perfect equality

The results show that Sector S_{36} has the minimum rate of inequality with 47.73% which means that this Sector is the one with the most minimal deviation from the uniform distribution in Sample 2.
The largest deviation from the perfect equality is in S_{12} with a rate of 97.28%, followed by S_{43},S_{23}, and S_{11}.

6.2.2 Evaluation Indicator cpUTR

The deviations in behaviour of the sectors of Sample 2 from the behaviour of the bivariate normal distribution are of further interest. Therefore again the scaled UTRs of all sectors are considered and compared with the $nUTR$ of the normal distribution.
Again 50 equidistant level cuts are considered. The level cuts are in the interval $[bl, max(N(0,0,10,10,0))]$ and thus $\alpha_j \in [1e-08, 0.0159]$, $j = 1, \ldots, 50$.

The scaled UTRs of the sectors and the $nUTR$ are shown in the following Figures 6.16 - 6.18

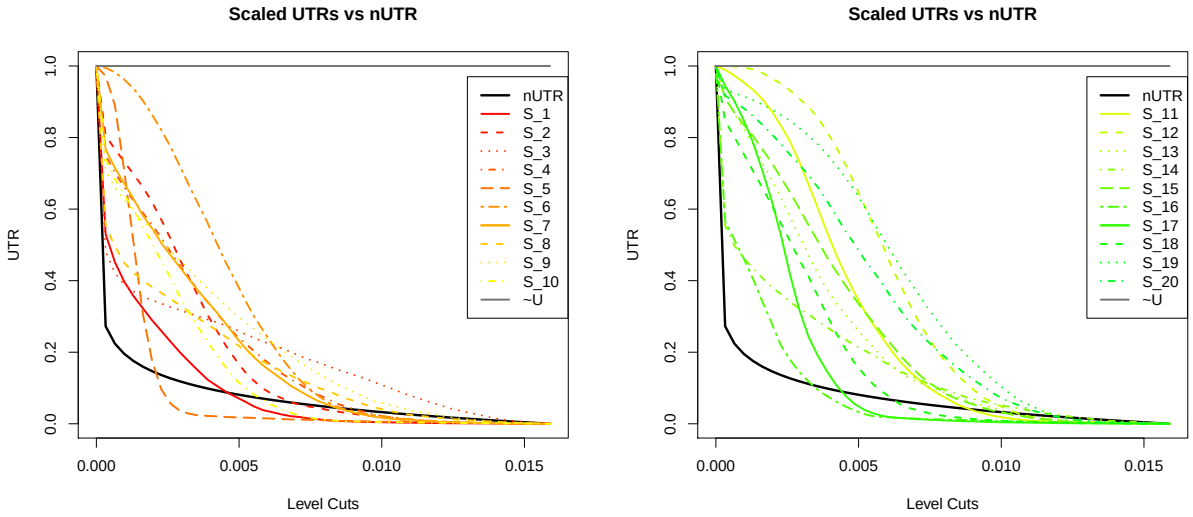

Figure 6.16: Scaled UTR for Sector S_1 to S_{10} and S_{11} to S_{20}

The scaled UTRs of the Sectors S_1 to S_{20} show a significant deviating behaviour than the UTR of the bivariate normal distribution.

From the next four figures in Figure 6.18, it can be seen that the behaviour of the scaled UTRs of the Sectors S_{28}, S_{37} and S_{53} seems similar to the $nUTR$. But the Sector S_{45} is the one with the most similar UTR to the UTR of the normal distribution.

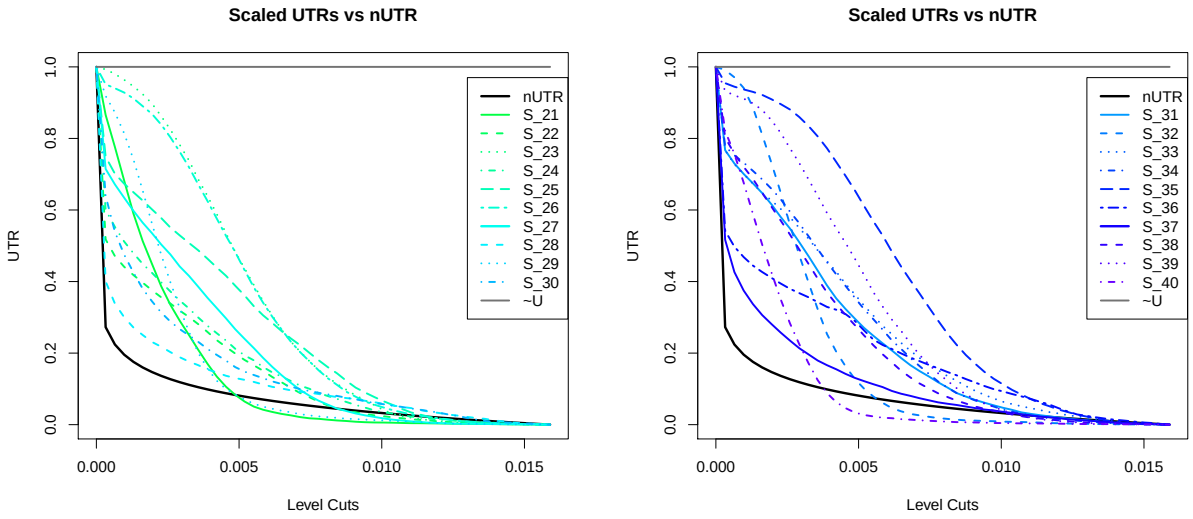

Figure 6.17: Scaled UTR for Sector S_{21} to S_{30} and S_{31} to S_{40}

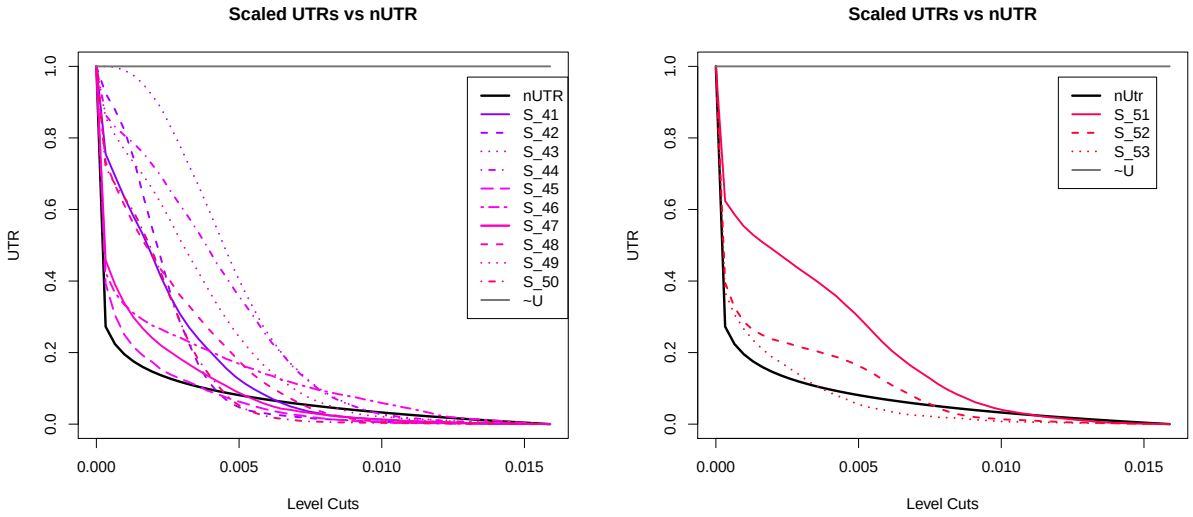

Figure 6.18: Scaled UTR for Sector S_{41} to S_{50} and S_{51} to S_{53}

The results for the second indicator $cpUTR$ are shown in Figure 6.19 below. The minimal deviation from the behaviour of $nUTR$ is in Sector S_8 with nearly 0.0002% and the maximal deviation is in S_5 with a rate of 25.57%. The rate for $cpUTR$ is in average about 8% and thus the deviation from the behaviour of the normal distribution is low for Sample 2.

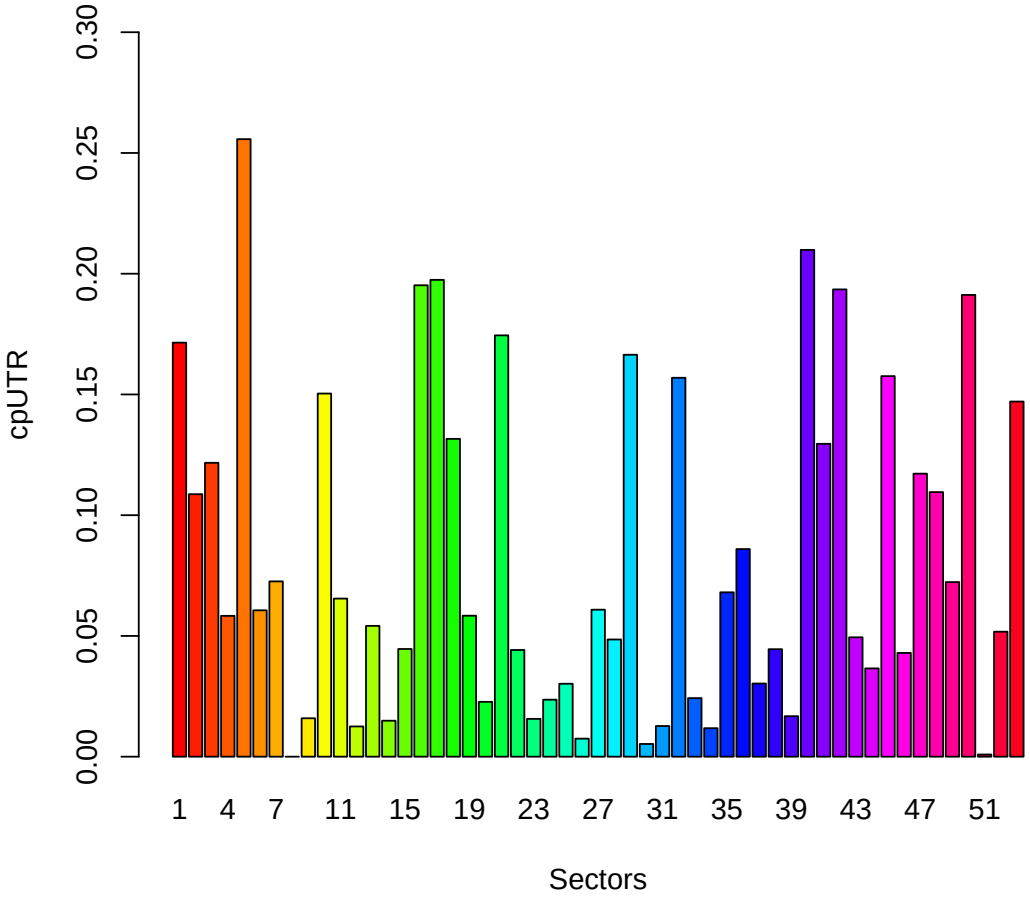

Figure 6.19: Indicator $cpUTR$ for Sample 2

6.2.3 Evaluation Indicator NGroups

In the following the results for the number of groupings within an evaluation area for 50 level cuts are presented. The equidistant level cuts are within the interval $[bl, ul]$ and hence

$\alpha_j \in [1.00e-08, 1.92e-06]$ with $j = 1, \ldots, 50$.

The indicator *NGroups* calculates the number of groupings per level within a defined coherent evaluation area. The same calculations like for Sample 1 are performed for Sample 2.

The next figures in Figure 6.20, show the obtained results, where it is clear that Sector S_{23} shows the maximal number of groupings within the evaluation area at the second level. It appears that the first 10 sectors do not have any groupings as from the 21th level cut, where in fact the sectors except Sector S_3 do not show any groupings as from level 15. The next ten Sectors S_i, with $i = 11, \ldots, 20$, show also a similar behaviour, since those do not have any groupings as from 8th level. The only exceptions are the Sectors S_{14}, which does not have any groupings in about as from the 18th level, and S_{16}, which does not have any as from the level 21.
Sectors S_i, $i = 21, \ldots, 30$, do not have any groupings as from the level 13, except S_{28} and S_{30}, which have at least one grouping in about up to the level 34 resp. up to the level 39. In the Sectors S_i, $i = 31, \ldots, 40$, are at least one grouping until the 13th level, as from then, only the Sectors S_{36} and S_{37} show groupings, where S_{36} has at least one grouping up to the 50th level.
The Sectors S_i, $i = 41, \ldots, 50$, show groupings up to the level 11, where S_{45}, S_{46} and S_{47} have at least one grouping until the 31th level. The Sector S_{53} show groupings up to the level 39.

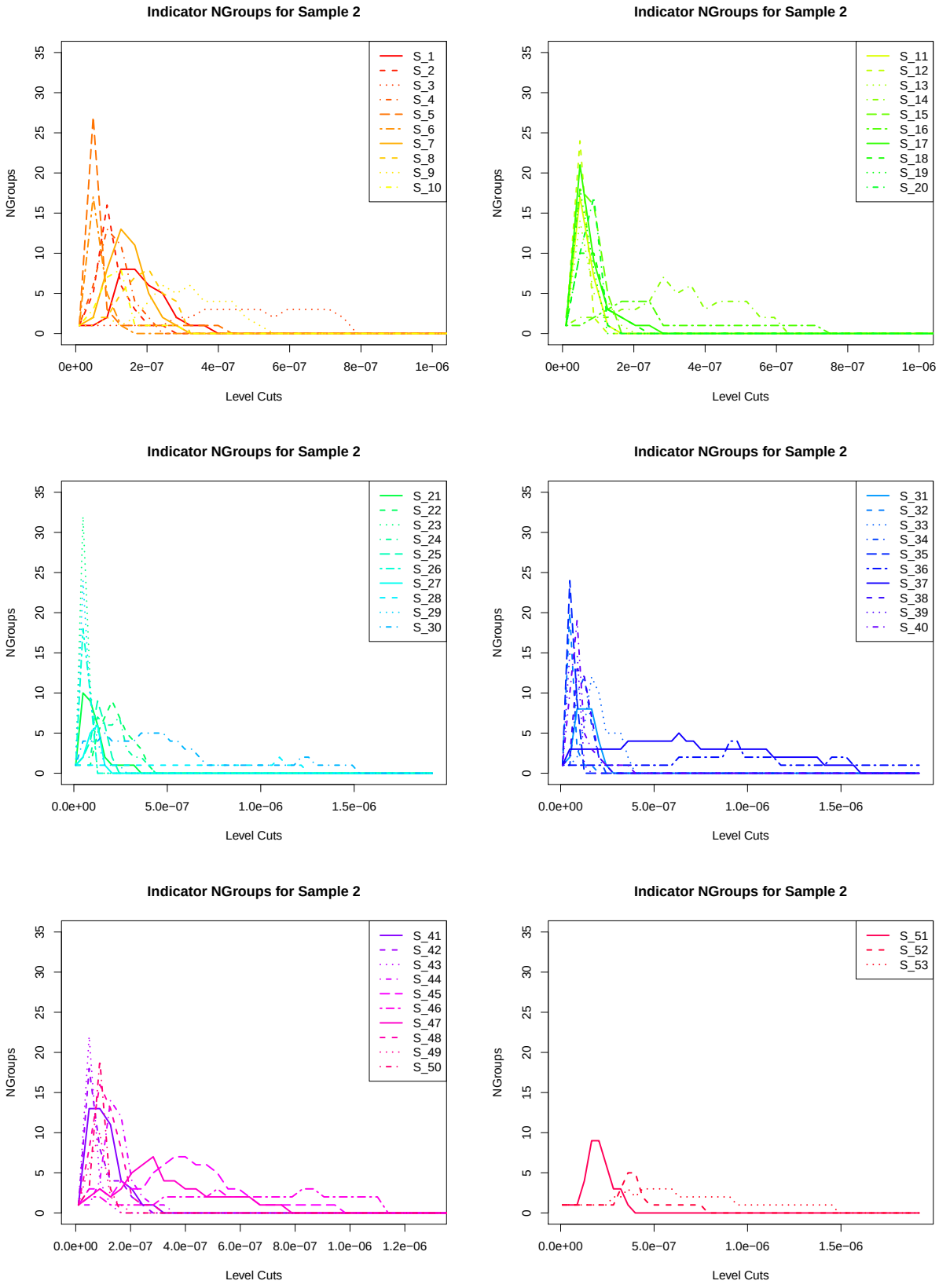

Figure 6.20: Indicator *NGroups* for Sector S_1 to S_{53}

6.2.4 Evaluation Indicator modCHI

In this last part of the evaluation of Sample 2 the separation of the groupings of the sectors is determined.
The same steps like for Sample 1 are done and since Sample 2 has many sectors, the evaluation area including the groupings is shown only for one sector.
Therefore, in Figure 6.21, Sector S_{23} including the maximum number of groupings at the second level is presented:

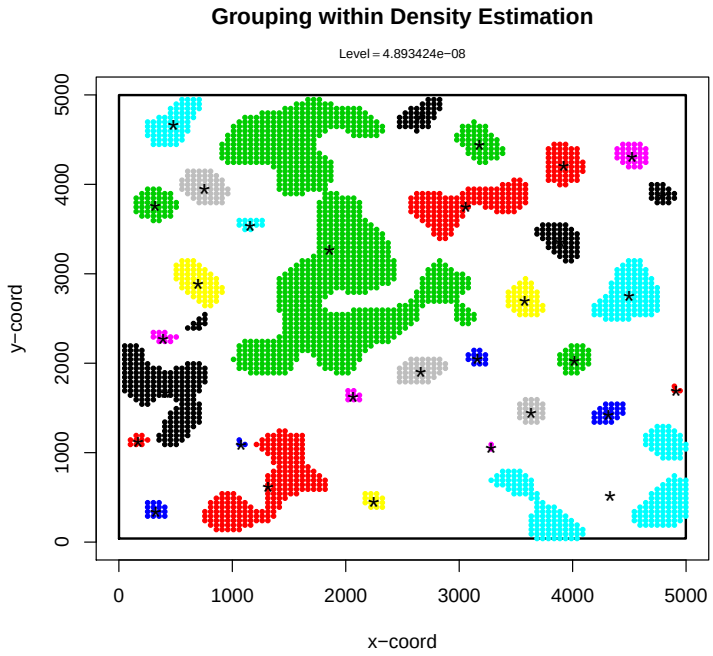

Figure 6.21: Groupings in Sector S_{23} of Sample 2 at the second level cut

The indicator $modCHI$ is calculated for all sectors of Sample 2 and is shown in the following figures in Figure 6.22, where $modCHI$ takes the value 0, if there is only one grouping in the eval.area at this level. The curves in these figures do not fall always towards 0, since if there is not any grouping within the eval.area at a level, the value for $modCHI$ becomes NA at this level.
It appears that at the second, but mostly at the third level the largest rates for the indicator $modCHI$ are obtained. Sector S_{23} has the maximum rate with 184.57. Other sectors with a high rate for $modCHI$ at these levels are S_5, S_{15}, S_{17} and S_{39}. Sectors with a low rate are among others S_{46}, S_{47} and S_{53}. There is a high concentration to one grouping in these sectors.

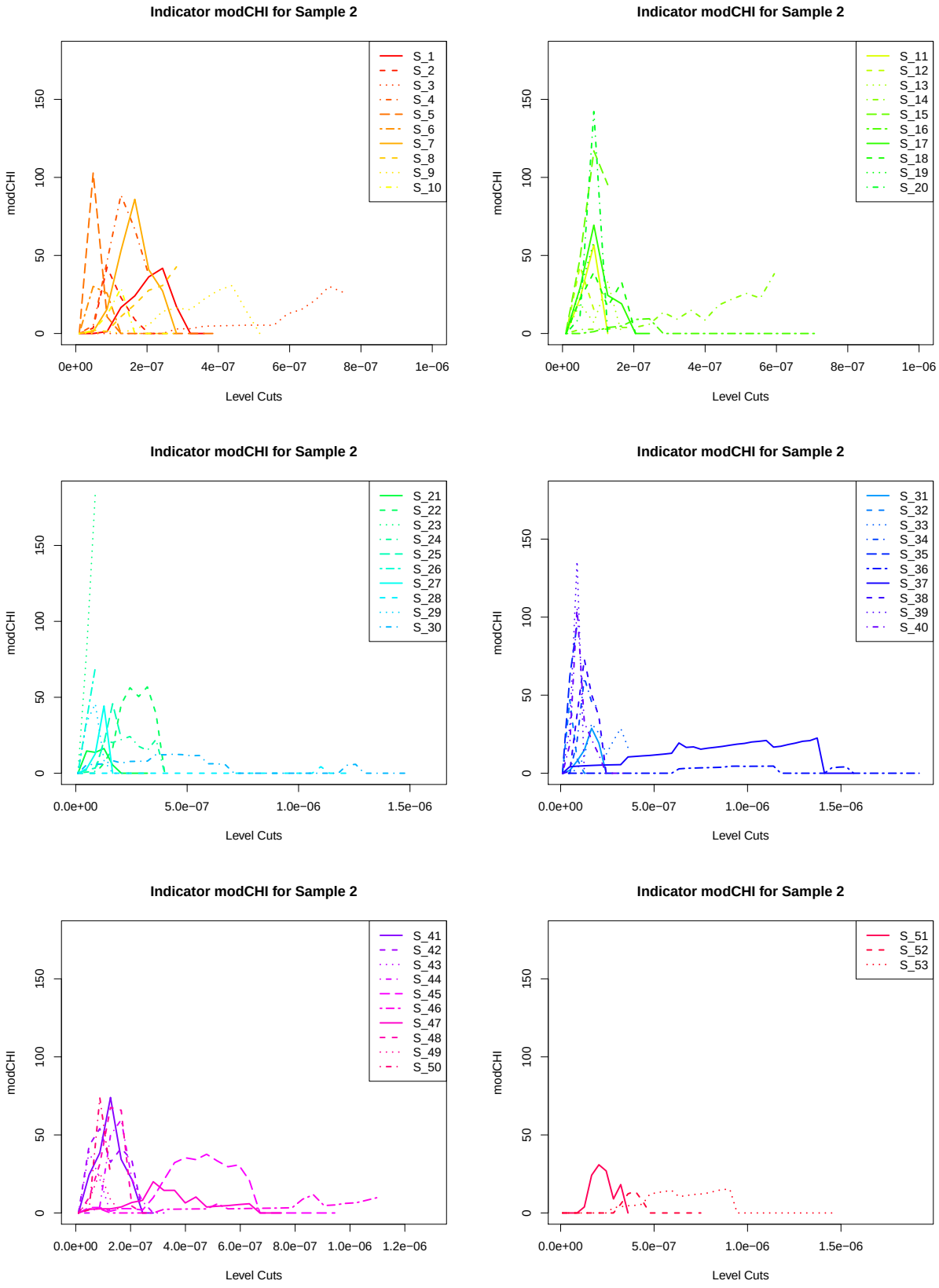

Figure 6.22: Indicator $modCHI$ for Sector S_1 to S_{53}

Chapter 7

Summary and Conclusion

This diploma thesis deals with the basic research of statistical support on the issue of computer-based assessment of pathological characteristics of brain tumors and it has been performed on request of a team of scientists from the Neurology department of the General Hospital Vienna (AKH).

The motivation of this work was the investigation of statistical methods which can be used as indicators for the computer-based analysis and assessment of cell activities in human brain tumors. The aim was the definition of some indicators which ensure an objective assessment. In this diploma thesis two digitalized human brain tumor tissue-sections were analyzed by using the free and open-source software environment **R**, version **2.14.2**.

At first the scanned and digitalized brain tumor samples underwent a process of segmentation in several parts and a process of determination of the marked cell nuclei during the preprocessing by DI Andreas Walser. This splitting into sectors was necessary because the digitalized slides had in average resolutions of the size of 100.000x100.000 pixels and the processing of such an image with a color depth of 24 Bit would need a memory consumption of 30 Gigabyte. But the images could also have larger sizes and this would yield to the need of huge memory consumptions, i.e. 100GB or larger. Hence, a special computer would be needed. Therefore the slides were divided into sectors of size 5000x5000 pixels.

The next step after the preprocessing was the performing of a two dimensional kernel density estimation with a Gaussian kernel of each sector, since the size of the sectors were still to large for further analysis.

During this thesis mainly two issues were of interest: On the one side, any information

about the properties of the measurements has been tried to accomplish. On the other side the spatial distribution of potential groupings has been considered.

For the first issue two indicators $giniUTR$ and $cpUTR$ have been defined as measures for inequality within the behaviour of the distribution of each sector. The idea was to consider the behaviour of the density estimation by considering points of intersection with the density along the z-axis. Thus the UTR, meaning **upper-to-total**- ratio, was defined, which calculates the ratio of the number of the values of the density estimation that are bigger than a level in relation to the total number of the values of the density estimation. By using the Gini-Index, which is a measure for the inequality, the first indicator was defined and gives the deviation of the UTRs from the UTR of the uniform distribution.
The first investigated Sample 1 delivered a maximum deviation with 96.08% for its Sector S_5 and the minimum deviation was in S_6 with a rate of 73.35%. The second Sample 2, which was much larger than the first one, showed a maximum deviation in its Sector S_{12} with a rate 97.28% and the minimum inequality was in S_{36} with 47.73%.

The indicator $cpUTR$ was defined as a measure for the comparison of the behaviour of the UTR of the sectors with the UTR of the normal distribution for several level cuts $\boldsymbol{\alpha} \in \mathbb{R}^m$ along the z-axis. For this purpose the Gini-Index again had been considered for both UTR of both distributions.
The obtained results showed that in Sample 1 the deviation from the normal distribution was varying between 1.19% and 10.2%. For Sample 2 the values were between 0.0002% and 25.57%.

For the second issue of interest, two other indicators were defined $NGroups$ and $modCHI$, which were supposed to give any information on the spatial distribution of possible groupings within the density estimation, for each level cut within a predetermined coherent evaluation area containing the majority of the density.
The indicator $NGroups$ was defined as the total number of the groupings within the evaluation area at each level and delivered for Sample 1, a maximal number of groupings for its Sector S_7 with a rate of 9 at the 4th level, and for the second Sample 2, a maximal value of 32 groupings for the Sector S_{23} at the 2nd level.

The last indicator $modCHI$ was supposed to measure the separation of these groupings within the density estimations. Therefore the setting of the density groupings in relation to the distances between those groupings has been considered. A modified version of the

Calinski Harabasz Index is an appropriate method to measure the dissimilarity between groupings over the similarity within groupings.
For Sample 1, the maximum separation was obtained for Sector S_{11} at the 13th level with a rate of 36.43, and for Sample 2, the Sector S_{23} showed the maximal separation with a rate of 184.57 at the 2nd level.

In conclusion, this diploma thesis demonstrated the usefulness of statistical methods for the computer-based assessment of human brain tumors. It requires further research on this topic and it is also reasonable to apply other statistical techniques. But nevertheless, the problem of the size of the digitalized original slides still exists, which will require the process of preprocessing and splitting the slides into sectors.
There also might be other approaches without the application of a kernel density estimation, i.e. splitting the sectors again into parts and may performing a regression analysis and modelling the counts of the data points per part. But there is further research needed to find the most appropriate statistical method.

In summary it can be said, that further research and improvement on this topic are required to define an automated software solution that is based on statistical techniques. This is beyond the scope of this thesis.

Appendix A

R-Code

A.1 data.read()

Every sector of both samples is imported separately into **R** and is saved in a list. Because of the structure of the received ASCII-files, it was necessary to write a separate **R**-function for importing the data. This function removes unnecessary information for further analysis and converts the structure of data into a data.frame.

```
1  ### Function for reading the Data after Preprocessing:
2  setGeneric("data.read", function(f,...){ standardGeneric("data.read") })
3  setMethod("data.read", definition=function(f){
4
5     dat<-scan(f, what="numeric",  sep=" ",
6     na.strings=c("Contour", "Number:","", "-->", "ContourArea:", "x:", "y:"))
7
8     datn<-as.numeric(dat)
9     ind<-which(!is.na(datn))
10
11    i<-seq(1,length(datn),1)
12    pp<-which((ind[i+1]-ind[i])==7)
13    x<-datn[ind[pp+1]]
14    pl<-which((ind[i+1]-ind[i])==3)
15    y<- datn[ind[pl+1]]
16    cc<-which((ind[i+1]-ind[i])==5)
17    cont<-c(datn[ind[cc]])
18
19    M<-cbind(x,y)
20    Mn<-cbind(x,y, NA)
21
22    j<-seq(1,(length(cc)-2),1)
23
24    con1<-rep(cont[1], trunc(cc[2]/2)-1, each=T)
25    con<-rep(cont[j+1], (trunc(cc[j+2]/2)-trunc(cc[j+1]/2)-1), each=T)
26
```

```
27    r<-(dim(Mn)[1]-(length(con1)+length(con)))
28    con2<-rep(cont[length(cont)], r, each=T)
29
30    Mn[1:length(con1),3]<-con1
31    Mn[(length(con1)+1):(length(con1)+length(con)),3] <- con
32    Mn[(length(con1)+length(con)+1):dim(Mn)[1],3]<-con2
33
34  return(Mn)
35 }
36 )
37
38 #daten<-list()
39 #f<-file("C:/.../out3.txt")
40 #dat<-data.read(f)
41 #daten[[1]]<-dat
42 #do this step for all data of both samples and save as list:
43 #save(file="C:\\...\\daten.RData", "daten")
44 #load("C:\\...\\daten.RData")
```

A.2 Bandwidth()

This function computes the optimal bandwidth for the kernel density estimation for all sectors of both samples.

```
1  ## Function for Determination of the optimal Bandwidth:
2  library(MASS)
3  setGeneric("Bandwidth", function(daten,...){ standardGeneric("Bandwidth") })
4  setMethod("Bandwidth", signature=c("list"), definition=function(daten){
5
6    BW<-matrix(ncol=2, nrow=length(daten))
7    colnames(BW)<-c("bwx","bwy")
8
9    for(i in 1:length(daten)){
10    BW[i,]<-c(bandwidth.nrd(daten[[i]][,1]),bandwidth.nrd(daten[[i]][,2]))
11   }
12   h<-c(mean(BW[,1]), mean(BW[,2]))
13
14  return(h)
15 }
16 )
17 #h<-Bandwidth(daten)
18 #save(file="C:\\...\\bw.RData", "h")
19 #load("C:\\...\\bw.RData")
```

A.3 kde2()

The bivariate kernel density estimation for all sectors is done via the following function:

```
## Function calculation the 2D density estimation for all data
## with the optimal bandwidth:
library(MASS)
setGeneric("kde2", function(daten,h,n){ standardGeneric("kde2") })
setMethod("kde2", definition=function(daten,h,n){

  dichte<-list()
  for(i in 1:length(daten)){
      dichte[[i]] <- kde2d(daten[[i]][,1], daten[[i]][,2], h=h, n=n)
  }

 return(dichte)
}
)
#dichten <- kde2(daten,h, 100)
#save(file="C:\\...\\dichten.RData", "dichten")
#load("C:\\...\\dichten.RData")
```

A.4 kde2dplot()

The function below is a function for plotting the kde2d, where the code is basing on the code from [Francois, 2011b].

```
## Function for plotting the 2D KDE:
kde2dplot <- function(d, cuts, plot2d=T,
# d is a 2d density computed by kde2D
 ncol=50,          # the number of colors to use
 zlim=c(0,max(z)), # limits in z coordinates
 nlevels=20,       # the number of colour levels
 theta=30,         # see option theta in persp
 phi=30)           # see option phi in persp
          {
  z <- d$z
  nrz <- nrow(z)
  ncz <- ncol(z)

  couleurs  <- tail(topo.colors(trunc(1.4 * ncol)),ncol)
  fcol      <- couleurs[trunc(z/zlim[2]*(ncol-1))+1]
  dim(fcol) <- c(nrz,ncz)
  fcol      <- fcol[-nrz,-ncz]
 if(!missing(cuts) & plot2d==T){
  par(mfrow=c(1,2),mar=c(0.5,0.5,0.5,0.5))
    res<-persp(d,col=fcol,zlim=zlim,theta=theta,phi=phi,xlab="x", ylab="y",
               zlab="density")
```

```
22     xx<-expand.grid(c(min(d$x), max(d$x)),c(min(d$y), max(d$y)))
23
24     polygon(trans3d(c(min(d$x), max(d$x),max(d$x), min(d$x),min(d$x)),
25                     c(min(d$y),min(d$y),max(d$y),max(d$y),min(d$y)),cuts,pmat=res),
26                      density=20, border=F, col="red")
27    par(new=T)
28    persp(d,col=fcol,zlim=zlim,theta=theta,phi=phi,xlab="x", ylab="y",
29          zlab="density")
30    #title(main="2D Density-Estimation", line=-1)
31    mtext(bquote(Level == .(cuts)), line=-0.8)
32    lines(trans3d(c(min(d$x), max(d$x),max(d$x), min(d$x),min(d$x)),
33                    c(min(d$y),min(d$y),max(d$y),max(d$y),min(d$y)),
34                     cuts,pmat=res),col="red", lty="dotted")
35
36    par(mar=c(2,2,2,2))
37    image(d,col=couleurs)
38    contour(d,add=T,levels=cuts)
39    box()
40  }
41  if(missing(cuts) & plot2d==T){
42    par(mfrow=c(1,2),mar=c(0.5,0.5,0.5,0.5))
43    persp(d,col=fcol,zlim=zlim,theta=theta,phi=phi,xlab="x", ylab="y",
44          zlab="density")
45    #title(main="2D Density-Estimation", line=-1)
46    par(mar=c(2,2,2,2))
47    image(d,col=couleurs)
48    contour(d,add=T,nlevels=nlevels)
49    box()
50  }
51  if(plot2d==F){
52    par(mar=c(2,2,2,2))
53    image(d,col=couleurs)
54    contour(d,add=T,nlevels=nlevels)
55  #  title(main="Contour Plot 2D-Density Estimation", line=1)
56    box()
57  }
58 }
59 setGeneric("kde2dplot")
```

A.5 baseline()

In the following the baseline of all sectors of both samples is determined:

```
1 ## Function for computing the baseline for all Samples:
2 setGeneric("baseline", function(dichte){ standardGeneric("baseline") })
3 setMethod("baseline", definition=function(dichte){
4
5   minLevel<-vector()
6   iso<-sapply(1:length(dichte), function(i) contourLines(dichte[[i]]))
7
8   for(j in 1:length(iso)){
```

```
    level<-sapply(1:length(iso[[j]]), function(x) iso[[j]][[x]]$level)
    minLevel[j]<-min(level)
  }
  minlevel<-min(minLevel)

 return(minlevel=minlevel)
}
)
```

A.6 gvt2d()

The bivariate uniform distribution is computed via the function below, where the code is taken from [Arminger, 2009], with a little modification.

```
## Function computing the theoretical bivariate uniform distribution
gvt2d<-function(x.vec = c(1,4),y.vec = c(2,5),n=100){
 n.axes=max(x.vec,y.vec)
 x <- seq(from=0, to=n.axes,length=n)
 y <- x
 z <- matrix(0, nrow=n, ncol=n)
 for (i in 1:n){
   for (j in 1:n){
     if(x.vec[1]<=x[i] && x[i]<=x.vec[2] && y.vec[1]<=y[j] && y[j]<=y.vec[2]){
       z[i,j] <- 1/((x.vec[2]-x.vec[1])*(y.vec[2]-y.vec[1]))
     }
   }
 }
return(list(x=x, y=y, z=z))
}
setGeneric("gvt2d")
```

A.7 nvt2d()

The function is for the computing of the bivariate normal distribution, where the code is taken from [Francois, 2011a] with a little modification.

```
## Function for computing the theoretical bivariate normal distribution
nvt2d<-function(x1,x2,
mu1=0, # setting the expected value of x1
mu2=0, # setting the expected value of x2
s11=1, # setting the variance of x1
s12=0, # setting the covariance between x1 and x2
s22=1, # setting the variance of x2
rho=0){
 # setting the correlation coefficient between x1 and x2
```

```
 f<-function(x1,x2){
    term1 <- 1/(2*pi*sqrt(s11*s22*(1-rho^2)))
    term2 <- -1/(2*(1-rho^2))
    term3 <- (x1-mu1)^2/s11
    term4 <- (x2-mu2)^2/s22
    term5 <- -2*rho*((x1-mu1)*(x2-mu2))/(sqrt(s11)*sqrt(s22))
    term1*exp(term2*(term3+term4-term5))
 }
# setting up the function of the multivariate normal density
 z<-outer(x1,x2,f) # calculating the density values
return(list(x=x1, y=x2, z=z))
}
setGeneric("nvt2d")
```

A.8 UTR()

The function below calculates the UTR and also the scaled $sUTR$, if there is a scaling value given as input for the level cuts.

```
setGeneric("UTR", function(d,minval,cuts,...){ standardGeneric("UTR") })
setMethod("UTR", definition=function(d,minval, cuts){

  baseline<-which(d$z> minval, arr.ind=TRUE)
  dznew<- d$z[baseline]
## calculates density > cuts, column corresponds to level cuts:
  res <- sapply(cuts, function(x) dznew >=  x)
  ## sum of the values >= cuts
  len <- apply(res, 2, sum)
## UTR:
 ratio.ges<-len/nrow(res)

 return(as.matrix(ratio.ges))
}
)
```

A.9 Inequ()

With the folowing function the first indicator $giniUTR$ is computed for all sectors and for all level cuts in one step.

```
## Function for computing Indikator 1:
library(ineq)
setGeneric("Inequ", function(dens,alpha,init,ndens,sc,...){ standardGeneric("Inequ") })
setMethod("Inequ",definition=function(dens,alpha,init,ndens,sc,...){
 ## minlevel=alpha[1]
```

```
6  ## init = initial sector to be analyzed (dens)
7  ## ndens= last sector to be analyzed (dens)
8  ## for the case of scaling with a factor:
9  if(!missing(sc)){
10   for(i in init:ndens){
11    fak<-sc/max(dens[[i]]$z)
12    dens[[i]]$z<-dens[[i]]$z*fak
13   }
14 }
15 else dens<-dens
16
17 IND1<-matrix(ncol=1, nrow=(ndens-init+1))
18 ## rows of UTRDd are the level cuts and columns are the sectors
19 UTRd<-matrix(nrow=length(alpha), ncol=(ndens-init+1))
20 vec<-seq(init, ndens,1)
21
22  for(i in 1:(ndens-init+1)){
23   d<-dens[[vec[i]]]
24   UTRd[,i]<- UTR(d, alpha[1], alpha)
25   IND1[i]<-ineq(UTRd[,i])
26  }
27
28 return(list(UTR=UTRd, Inequ=IND1))
29 }
30 )
```

A.10 nUTR()

This function delivers the analytically computed UTR for the bivariate normal distribution with $\mu_1 = \mu_2 = 0$ and $\sigma_1 = \sigma_2 = \sigma$ and $\rho = 0$.

```
1  ## Function for the analytical computation of the UTR for the
2  ## bivariate normal distribution with mu=0, rho=0, and sigma_1=sigma_2:
3  setGeneric("nUTR", function(sigma,nalpha,...){ standardGeneric("nUTR") })
4  setMethod("nUTR", definition=function(sigma, nalpha,...){
5    minlevel<-nalpha[1]
6    n<-length(nalpha)
7    maxtnvt<-nalpha[n]
8  ## Calculation of the level cuts as function
9    k<- function(i) minlevel+(i-1)*(maxtnvt-minlevel)/(n-1)
10 ## Calculation of the nUTR:
11   V<-function(i) log(2*pi*sigma*k(i))/log(2*pi*sigma*k(1))
12   nutr<-V(1:n)
13
14 return(nUTR=nutr)
15 }
16 )
```

A.11 CompNdist()

In the function below the second indicator *cpUTR* is computed.

```
## Function for Indicator 2 CompNdist:
setGeneric("CompNdist", function(dens, nalpha, init, ndens, sigma, maxtnvt)
  {standardGeneric("CompNdist")})
setMethod("CompNdist",
   definition=function(dens, nalpha, init, ndens, sigma, maxtnvt){

   n.utr<-nUTR(sigma, nalpha)
   ## Scaled UTR:
   UTRd<-Inequ(dens,nalpha,init,ndens,maxtnvt)$UTR
   ## Scaled Indicator Inequ
   IND1scaled<-Inequ(dens,nalpha,init,ndens,maxtnvt)$Inequ
   ## Difference between nUTR and Scaled UTRs:
   IND2<- abs(ineq(n.utr)-IND1scaled)

 return(list(sUTR=UTRd, CNdist=IND2))
}
)
```

A.12 dmat()

This function is supposed to convert the the list of the density estimation, which consists of 3 elements, where the first two elements are x- and y- coordinates of the grid points where the values are estimated and the third element z is the matrix of the values of the density estimation. As result a matrix of the density estimation is obtained.
This function will be needed later in the function `grouping()`.

```
## Function for unlisting and converting the list of the densities for all
## sectors of both samples, where the density values < minlevel become 0:
setGeneric("dmat", function(d, minlevel){standardGeneric("dmat")})
setMethod("dmat", signature=c("list","numeric"),
definition=function(d, minlevel){
  coord<-as.matrix(expand.grid(d$x,d$y))
  n<-length(d$x)*length(d$y)
  ss<-seq(1,n,1)
  MM<-cbind(coord[ss,], d$z[ss])
  colnames(MM)<-c("x","y","z")
  ind<-which(MM[,3] < minlevel)
  MM[,3][ind]<-0

 return(MM)
}
)
```

A.13 Areanew()

With the function below, the evaluation area for a sector for the computing of the groupings is determined.

```
## Function for defining the eval.area:
setGeneric("Areanew", function(Area, K){standardGeneric("Areanew")})
setMethod( "Areanew", signature=c("matrix", "matrix"),
definition= function(Area, K){

## 1.) ++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++ ##
## separation of the image into 4 quadrants:
qu1<-cbind(c(5000/2, 5000/2, 5000, 5000, 5000/2),
              c(5000/2, 5000, 5000,5000/2, 5000/2))
qu2<-cbind(c(1, 1, 5000/2, 5000/2,1),
             c(5000/2, 5000, 5000,5000/2, 5000/2))
qu3<- cbind(c(1, 1, 5000/2, 5000/2,1),
              c(1, 5000/2,5000/2,1, 1))
qu4<-cbind(c(5000/2, 5000/2, 5000, 5000,5000/2),
            c(1, 5000/2, 5000/2,1, 1))

## testing the density points, to which quadrant they belong
test.qu1<-point.in.polygon(K[,1], K[,2], qu1[,1], qu1[,2])
test.qu2<-point.in.polygon(K[,1], K[,2], qu2[,1], qu2[,2])
test.qu3<-point.in.polygon(K[,1], K[,2], qu3[,1], qu3[,2])
test.qu4<-point.in.polygon(K[,1], K[,2], qu4[,1], qu4[,2])

lqu1<-length(which(test.qu1>0))
lqu2<-length(which(test.qu2>0))
lqu3<-length(which(test.qu3>0))
lqu4<-length(which(test.qu4>0))

## testing the Area to which quadrant they belong:
Aqu1<-point.in.polygon(c(Area[1,1], Area[nrow(Area),1]),
   c(Area[1,2], Area[nrow(Area),2]), qu1[,1], qu1[,2])
Aqu2<-point.in.polygon(c(Area[1,1], Area[nrow(Area),1]),
   c(Area[1,2], Area[nrow(Area),2]), qu2[,1], qu2[,2])
Aqu3<-point.in.polygon(c(Area[1,1], Area[nrow(Area),1]),
   c(Area[1,2], Area[nrow(Area),2]), qu3[,1], qu3[,2])
Aqu4<-point.in.polygon(c(Area[1,1], Area[nrow(Area),1]),
   c(Area[1,2], Area[nrow(Area),2]), qu4[,1], qu4[,2])

## ++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++ ##

# 2.) Testing the image (Density):
### ********************************************************************** ###
# a) mainly right top:
if((lqu1 > lqu2) & (lqu1 > lqu3) & (lqu1 > lqu4) & ((Aqu4[1]>0 & Aqu2[2]>0) ||
   (Aqu1[1]>0 & Aqu2[2]>0))){
    if(  (Aqu4[1]>0 & Aqu2[2]>0) & (min(K[,2])==Area[1,2])){
    areanew<-cbind(c(Area[,1],Area[nrow(Area),1],Area[nrow(Area),1],
                max(K[,1]), max(K[,1])),
                c(Area[,2],Area[nrow(Area),2],max(K[,2]),max(K[,2]),min(K[,2])))
```

```
49      }
50      if((Aqu1[1]>0 & Aqu2[2]>0)||(Aqu4[1]>0&Aqu2[2]>0)&(min(K[,2])!=Area[1,2])){
51      areanew<-cbind(c(Area[,1], Area[nrow(Area),1],Area[nrow(Area),1],
52                  max(K[,1]),Area[1,1]),
53                  c(Area[,2], Area[nrow(Area),2],max(K[,2]),max(K[,2]),Area[1,2]))
54      }
55  }
56
57  # b) mainly right bottom:
58  if((lqu4 > lqu1) & (lqu4 > lqu2) & (lqu4 > lqu3) & (Aqu3[1]>0 & Aqu1[2]>0)){
59     if(Area[nrow(Area),2]==max(K[,2])){
60        areanew<-cbind(c(Area[,1],Area[nrow(Area),1],max(K[,1]),max(K[,1]),
61                         Area[nrow(Area):1,1] ),
62                       c(Area[,2],Area[nrow(Area),2],max(Area[,2]),min(K[,2]),
63                         rep(min(K[,2]), nrow(Area))))
64     }
65
66    if(Area[nrow(Area),2]!=max(K[,2])){
67       areanew<-cbind(c(Area[,1], Area[nrow(Area),1] ,max(K[,1]), Area[1,1]),
68                      c(Area[,2], Area[nrow(Area),2], min(K[,2]), Area[1,2]))
69    }
70  }
71  # c) right side:
72  if((((lqu4>lqu3)&(lqu1>lqu2))||(lqu4>lqu2)&(lqu1>lqu4))&((Aqu1[1]>0&Aqu4[2]>0)
73      ||(Aqu4[1]>0 & Aqu1[2]>0)||(Aqu3[1]>0 & Aqu2[2]>0))){
74   if((Aqu4[1]>0 & Aqu1[2]>0) ||(Aqu3[1]>0 & Aqu2[2]>0)){
75    areanew<-cbind(c(Area[,1],Area[nrow(Area),1],max(K[,1]),max(K[,1]),Area[1,1]),
76                   c(Area[,2],Area[nrow(Area),2],max(K[,2]),min(K[,2]),min(K[,2])))
77   }
78   if((Aqu4[1]>0 & Aqu1[2]>0) & (Area[nrow(Area),2]!=max(K[,2]))){
79    areanew<-cbind(c(Area[,1],Area[nrow(Area),1],Area[nrow(Area),1],Area[1,1]),
80                    c(Area[,2], Area[nrow(Area),2],min(K[,2]),min(K[,2])))
81   }
82
83   if(Aqu1[1]>0 & Aqu4[2]>0){
84    areanew<-cbind(c(Area[,1],Area[nrow(Area),1],max(K[,1]),max(K[,1]),Area[1,1]),
85                   c(Area[,2],Area[nrow(Area),2],min(K[,2]),max(K[,2]),max(K[,2])))
86
87   }
88  }
89  # d) middle, mainly right:
90  if((lqu3 >= lqu4) & (lqu1>lqu2) & (Aqu3[1]>0 & Aqu1[2]>0)){
91    areanew<-cbind(c(Area[,1],Area[nrow(Area),1],max(K[,1]),max(K[,1]),
92                     Area[nrow(Area):1,1] ),
93                    c(Area[,2],Area[nrow(Area),2],max(Area[,2]),min(K[,2]),
94                      rep(min(K[,2]), nrow(Area))))
95  }
96  # e) middle bottom:
97  if((((lqu4 > lqu1)&(lqu3 > lqu2))||(lqu4>lqu1 & lqu4>lqu3))&
98    ((Aqu2[1]>0 & Aqu1[2] >0)||(Aqu3[1]>0 & Aqu4[2]>0)||(Aqu4[1]>0 & Aqu2[2]>0)||
99     (Aqu2[1]>0 & Aqu4[2]>0))){
100   if(Aqu4[1]>0 & Aqu2[2]>0){
101     areanew<-cbind(c(Area[,1], Area[nrow(Area),1],min(K[,1]),min(K[,1]),
102                      Area[1,1], Area[1,1]),
103                    c(Area[,2], Area[nrow(Area),2], max(Area[,2]),
```

```
                      min(K[,2]), min(K[,2]), Area[1,2]))
  }
  if(!(Aqu4[1]>0 & Aqu2[2]>0)){
    areanew<-cbind(c(Area[,1],Area[nrow(Area),1],Area[nrow(Area):1,1],Area[1,1]),
            c(Area[,2],Area[nrow(Area),2],rep(min(K[,2]),nrow(Area)),Area[1,2]))
  }
}
# f) mainly left bottom:
if( (lqu3 > lqu4) & (lqu3> lqu2) & (lqu3 > lqu1) & ((Aqu2[1]>0 & Aqu3[2]>0))){
  if(Area[1,1]<=Area[nrow(Area),1]){
    areanew<-cbind(c(Area[,1],Area[nrow(Area),1],Area[nrow(Area),1],min(K[,1])),
                   c(Area[,2],Area[nrow(Area),2],min(K[,2]),min(K[,2])))
  }
  if(Area[1,1]>Area[nrow(Area),1]){
     areanew<-cbind(c(Area[,1],Area[nrow(Area),1],Area[nrow(Area),1],Area[1,1]),
                    c(Area[,2],Area[nrow(Area),2],max(K[,2]),max(K[,2])))
  }
}
# g) left side:
if((lqu2 > lqu1)&(lqu3 > lqu4)&((Aqu4[1]>0 & Aqu2[2]>0)||(Aqu3[1]>0 & Aqu2[2]>0)
   ||( Aqu3[1]>0 & Aqu1[2]>0)||(Aqu4[1]>0 & Aqu1[2]>0))){
  if(Aqu4[1]>0 & Aqu1[2]>0||(Aqu3[1]>0 & Aqu2[2]>0)||(Aqu3[1]>0 & Aqu1[2]>0)){
   areanew<-cbind(c(Area[,1],Area[nrow(Area),1],Area[nrow(Area),1],min(K[,1]),
                    min(K[,1]),Area[1,1]),
                  c(Area[,2],Area[nrow(Area),2],max(K[,2]),max(K[,2]),
                    min(K[,2]),min(K[,2])))
  }
 if(!(Aqu4[1]>0&Aqu1[2]>0)& !(Aqu3[1]>0 & Aqu2[2]>0)& !(Aqu3[1]>0 & Aqu1[2]>0)){
  areanew<-cbind(c(Area[,1], Area[nrow(Area),1] ,Area[nrow(Area),1],
                   min(K[,1]), min(K[,1]),Area[nrow(Area):1,1], Area[1,1] ),
                 c(Area[,2], Area[nrow(Area),2], max(K[,2]), max(K[,2]),
                   min(K[,2]),rep(min(K[,2]), nrow(Area)), Area[1,2]))
 }
}
# h) special case left middle:
if( (lqu4 > lqu3) & (lqu2>lqu1) & (lqu3 > lqu2) & (Aqu1[1]>0 & Aqu1[2]>0)){
  areanew<-cbind(c(Area[,1], Area[nrow(Area),1], Area[nrow(Area),1],
                   min(K[,1]), min(K[,1]),Area[1,1]),
                 c(Area[,2], Area[nrow(Area),2], max(K[,2]), max(K[,2]),
                   min(K[,2]), min(K[,2])))
}
# i) mainly left top:
if((lqu2 > lqu1) & (lqu2 > lqu3) & (lqu2 > lqu4) & ((Aqu1[1]>0 & Aqu3[2]>0)
 ||(Aqu2[1]>0 & Aqu3[2]>0)||(Aqu3[1]>0 & Aqu2[2]>0)||(Aqu1[1]>0 & Aqu2[2]>0)
 ||( Aqu3[1]>0 & Aqu1[2]>0)||(Aqu4[1]>0 & Aqu1[2]>0))){
  if((Area[1,1]<=Area[nrow(Area),1 ]) & (Aqu3[1]>0 & Aqu2[2]>0)){
  areanew<-cbind(c(Area[,1],Area[nrow(Area),1],min(K[,1]),min(K[,1]),Area[1,1]),
                 c(Area[,2],max(K[,2]),max(K[,2]),min(K[,2]),Area[1,2]))
  }
  if((Area[1,1]<=Area[nrow(Area),1 ]) & ( Aqu2[1]>0 & Aqu3[2]>0)){
    areanew<-cbind(c(Area[,1],Area[nrow(Area),1],min(K[,1]),min(K[,1])),
                   c(Area[,2],min(K[,2]),min(K[,2]),max(K[,2])))
  }
  if((Area[1,1]<=Area[nrow(Area),1]) &
     ((Aqu3[1]>0 & Aqu1[2]>0)||(Aqu4[1]>0 & Aqu1[2]>0))){
```

```
159      areanew<-cbind(c(Area[,1],Area[nrow(Area),1],Area[nrow(Area),1],
160                       min(K[,1]),min(K[,1]) ),
161                     c(Area[,2],Area[nrow(Area),2],max(K[,2]),
162                       max(K[,2]),Area[1,2]))
163    }
164    if((Area[1,1] > Area[nrow(Area),1 ]) &
165        (!((Area[1,1]<=Area[nrow(Area),1 ]) & ( Aqu2[1]>0 & Aqu3[2]>0 ||
166           (Aqu3[1]>0 & Aqu2[2]>0))) || !((Area[1,1]<=Area[nrow(Area),1 ]) &
167            (( Aqu3[1]>0 & Aqu1[2]>0) || (Aqu4[1]>0 & Aqu1[2]>0))))){
168      if(!(Aqu3[1]>0 & Aqu2[2]>0)){
169       areanew<-cbind(c(Area[,1],Area[nrow(Area),1],Area[nrow(Area),1],Area[1,1]),
170                      c(Area[,2],Area[nrow(Area),2],max(K[,2]),max(K[,2])))
171      }
172      if((Aqu3[1]>0 & Aqu2[2]>0)){
173       areanew<-cbind(c(Area[,1],Area[nrow(Area),1],Area[nrow(Area),1],
174                        min(K[,1]),min(K[,1]),Area[nrow(Area):1,1] ),
175                      c(Area[,2],Area[nrow(Area),2],max(K[,2]),max(K[,2]),
176                        min(K[,2]),rep(min(K[,2]),nrow(Area))))
177      }
178    }
179  }
180  # j) special case left top:
181  if(max(lqu1,lqu2,lqu3,lqu4)==lqu2 & (Aqu2[1]>0 & Aqu2[2]>0) &
182        (Area[1,2]!=Area[nrow(Area),2])){
183    if(Area[nrow(Area),1] <= Area[1,1]){
184      areanew<-cbind(c(Area[,1],Area[nrow(Area),1],Area[nrow(Area):1,1]),
185                     c(Area[,2],Area[nrow(Area),2],rep(max(K[,2]),nrow(Area))))
186     }
187    if(!(Area[nrow(Area),1] <= Area[1,1])){
188      areanew<-cbind(c(Area[,1],Area[nrow(Area),1],Area[nrow(Area):1,1],
189                       min(K[,1]),min(K[,1]),Area[1,1] ),
190                     c(Area[,2],Area[nrow(Area),2],rep(max(K[,2]),nrow(Area)),
191                       max(K[,2]),min(K[,2]),Area[1,2]))
192    }
193  }
194  # k) middle top:
195  if((lqu1>lqu4)&(lqu2>lqu3)&(Aqu1[1]>0 & Aqu3[2]>0)){
196    areanew<-cbind(c(Area[,1],Area[nrow(Area),1],Area[nrow(Area),1],
197                     max(K[,1]),Area[1,1]),
198                   c(Area[,2],Area[nrow(Area),2],max(K[,2]),
199                     max(K[,2]),Area[1,2]))
200  }
201  # l) special case left bottom:
202  if((Aqu3[1]>0 & Aqu3[2]>0) & (!isTRUE(all.equal(lqu1,lqu2, tol=0.15)) ||
203    !isTRUE(all.equal(lqu1,lqu3, tol=0.15)) ||
204    !isTRUE(all.equal(lqu1,lqu4, tol=0.15))) & (Area[nrow(Area),2]!=Area[1,2])){
205   if(Area[nrow(Area),1]==Area[1,1]){
206    areanew<-cbind(c(Area[,1],Area[nrow(Area),1]),c(Area[,2],Area[nrow(Area),2]))
207   }
208   if(Area[nrow(Area),1]!=Area[1,1]){
209     areanew<-cbind(c(Area[,1],Area[nrow(Area),1],min(K[,1]),min(K[,1])),
210                    c(Area[,2],Area[nrow(Area),2],min(K[,2]),Area[1,1]))
211   }
212  }
213  # m) special case: only in the third quadrant
```

```
214 if((Aqu3[1]>0 & Aqu3[2]>0) & (!isTRUE(all.equal(lqu1,lqu2, tol=0.15)) ||
215   !isTRUE(all.equal(lqu1,lqu3, tol=0.15)) ||
216   !isTRUE(all.equal(lqu1,lqu4, tol=0.15))) & (Area[nrow(Area),2]==Area[1,2])
217   & any((isTRUE(all.equal(lqu1,lqu2, tol=0.15)) & (lqu1>0 | lqu2>0)),
218   (isTRUE(all.equal(lqu1,lqu3, tol=0.15)) & (lqu1>0 | lqu3>0)),
219   (isTRUE(all.equal(lqu1,lqu4, tol=0.15)) & (lqu1>0 | lqu4>0)))){
220
221    areanew<-cbind(c(min(K[,1]),min(K[,1]),max(K[,1]),max(K[,1]),min(K[,1])),
222                   c(min(K[,2]),max(K[,2]),max(K[,2]),min(K[,2]),min(K[,2])))
223 }
224 # n) special case: only in the third quadrant
225 if((Aqu3[1]>0 & Aqu3[2]>0) & (!isTRUE(all.equal(lqu1,lqu2, tol=0.15)) ||
226    !isTRUE(all.equal(lqu1,lqu3, tol=0.15)) ||
227    !isTRUE(all.equal(lqu1,lqu4, tol=0.15))) & (Area[nrow(Area),2]==Area[1,2])
228    & (!any((isTRUE(all.equal(lqu1,lqu2, tol=0.15)) & (lqu1>0 | lqu2>0)),
229    (isTRUE(all.equal(lqu1,lqu3, tol=0.15)) & (lqu1>0 | lqu3>0)),
230    (isTRUE(all.equal(lqu1,lqu4, tol=0.15)) & (lqu1>0 | lqu4>0))))){
231
232      areanew<-cbind(c(Area[,1],Area[nrow(Area),1],Area[nrow(Area):1,1]),
233                     c(Area[,2],Area[nrow(Area),2],rep(min(K[,2]),nrow(Area))))
234 }
235 # o) special case: uniformly distributed with a concentratin to right top
236 if(isTRUE(all.equal(lqu1,lqu2,tol=0.15)) & isTRUE(all.equal(lqu1,lqu3,tol=0.15))
237    & isTRUE(all.equal(lqu1,lqu4, tol=0.15)) & ((Aqu1[1]>0 & Aqu1[2]>0) ||
238     (Aqu2[1]>0 & Aqu2[2]>0) || (Aqu3[1]>0 & Aqu3[2]>0) ||
239     (Aqu4[1]>0 & Aqu4[2]>0) || (Aqu3[1]>0 & Aqu2[2]>0))){
240
241    areanew<-cbind(c(min(K[,1]),min(K[,1]),max(K[,1]),max(K[,1]),min(K[,1])),
242                   c(min(K[,2]),max(K[,2]),max(K[,2]),min(K[,2]),min(K[,2])))
243 }
244 # p)special case: uniformly distributed with a concentratin to right bottom
245 if((lqu3 > lqu1) & (lqu4> lqu2) & (lqu3 > lqu2) & (lqu1>lqu2) &
246    ((Aqu2[1]>0 & Aqu2[2]>0))){
247   areanew<-cbind(c(Area[,1],Area[nrow(Area),1],max(K[,1]),max(K[,1]),
248                    Area[nrow(Area):1,1]),
249                  c(Area[,2],Area[nrow(Area),2],max(Area[,2]),min(K[,2]),
250                    rep(min(K[,2]),nrow(Area))))
251 }
252 # q) special case: only in the 4th quadrant
253 if((Aqu4[1]>0 & Aqu4[2]>0) & (!isTRUE(all.equal(lqu1,lqu2, tol=0.15)) ||
254    !isTRUE(all.equal(lqu1,lqu3, tol=0.15)) ||
255    !isTRUE(all.equal(lqu1,lqu4, tol=0.15)))){
256
257   areanew<-cbind(c(Area[,1],Area[nrow(Area),1],Area[nrow(Area):1,1] ),
258                  c(Area[,2],Area[nrow(Area),2],rep(min(K[,2]),nrow(Area))))
259 }
260 # r) special case: only in the 2nd quadrant
261 if( (Aqu2[1]>0 & Aqu2[2]>0) & (!isTRUE(all.equal(lqu1,lqu2, tol=0.15)) ||
262    !isTRUE(all.equal(lqu1,lqu3, tol=0.15)) ||
263    !isTRUE(all.equal(lqu1,lqu4, tol=0.15))) &
264    !((lqu3 > lqu1) & (lqu4> lqu2) & (lqu3 > lqu2) & (lqu1>lqu2)) &
265    !(max(lqu1,lqu2,lqu3,lqu4)==lqu2 & (Area[1,2]!=Area[nrow(Area),2]))){
266
267     areanew<-cbind(c(Area[,1],Area[nrow(Area),1],Area[nrow(Area):1,1]),
268                    c(Area[,2],Area[nrow(Area),2],rep(max(K[,2]),nrow(Area))))
```

```
}
# s) special case: only in the first quadrant:
if((Aqu1[1]>0 & Aqu1[2]>0) & !((lqu4 > lqu3) & (lqu2>lqu1) & (lqu3 > lqu2)) &
  (!isTRUE(all.equal(lqu1,lqu2,tol=0.15)) ||
   !isTRUE(all.equal(lqu1,lqu3,tol=0.15)) ||
   !isTRUE(all.equal(lqu1,lqu4,tol=0.15)))){

   areanew<-cbind(c(Area[,1],Area[nrow(Area),1],Area[nrow(Area):1,1]),
                  c(Area[,2],Area[nrow(Area),2],rep(max(K[,2]),nrow(Area))))
}
# t) special case: middle bottom, concave curvature:
if((Aqu3[1]>0 & Aqu4[2]>0) & !((lqu4 > lqu3) & (lqu2>lqu1) & (lqu3 > lqu2)) &
  !(((lqu4 > lqu1) & (lqu3 > lqu2)) || (lqu4>lqu1 & lqu4>lqu3)) &
  (!isTRUE(all.equal(lqu1,lqu2, tol=0.15)) ||
   !isTRUE(all.equal(lqu1,lqu3, tol=0.15)) ||
   !isTRUE(all.equal(lqu1,lqu4, tol=0.15)))){

   areanew<-cbind(c(min(K[,1]),min(K[,1]),max(K[,1]),max(K[,1]),min(K[,1])),
                  c(min(K[,2]),max(K[,2]), max(K[,2]), min(K[,2]),min(K[,2])))
}

  return(Areanew=areanew)
}
)
```

A.14 groups.fix()

This function calculates for a sector the contour lines of possible groups at a given level.

```
## Function for determining contours of possible groups:
library(grDevices)
setGeneric("groups.fix", function(d, fix){ standardGeneric("groups.fix") })
setMethod("groups.fix", definition=function(d, fix){
  isofix<-contourLines(d, nlevels=1, levels=fix)
  if(length(isofix)==0) MAT<-cbind(0,0,0)
  if(length(isofix)==1) MAT<-cbind(isofix[[1]]$x, isofix[[1]]$y, 1)
  if(length(isofix)>1){
   fixmat<-lapply(1:length(isofix),
            function(x) cbind(isofix[[x]]$x, isofix[[x]]$y, x))
 ## converting list of density into matrix:
  nr<-sapply(1:length(fixmat), function(x) nrow(fixmat[[x]]))
  gr<-cumsum(nr)
  MAT<-matrix(nrow=sum(nr), ncol=3)
  MAT[1:gr[1],]<-fixmat[[1]]
  k<-seq(2,length(nr), 1)
  l<-1
   for(i in k){
     MAT[(gr[i-1]+1):gr[i], ]<-fixmat[[i]]
     l=l+1
   }
 }
```

```
 colnames(MAT)<-c("x","y", "group")
 return(MAT )
}
)
```

A.15 grouping()

In the function below the groupings within the evaluation area, determined via `Areanew()`, are computed the ouput is a list, containing as first element the matrix of the coordinates of the denisty points and in the third column the corresponding classification of the grouping. The second element of the output is the eval.area.

```
## Function determines the groupings within density per fix level cut:
#library(sp)
library(gstat)
setGeneric("grouping",function(d, minlevel, cuts, fix.value)
     {standardGeneric("grouping")})
setMethod("grouping", signature=c("list","numeric", "numeric", "numeric"),
    definition= function(d, minlevel, cuts, fix.value){

bl<- which(d$z> minlevel, arr.ind=TRUE)
M<-matrix(0, nrow=100, ncol=100)
M[bl]<-d$z[bl]
## consider density > minlevel
dnew<-list(x=d$x, y=d$y, z=M)

## determination of the possible Area for the evaluation:
borders<- contourLines(dnew, levels=minlevel)
## maximal contour is chosen:
geb<-sapply(1:length(borders), function(x) length(borders[[x]]$x))
nr<-which(geb==max(geb))
## possible evaluation area:
Area<-cbind(borders[[nr]]$x, borders[[nr]]$y)
## list of density converted into matrix:
dm<-dmat(dnew, minlevel)
##### ************************************************************ ######
K<- dm[dm[,3]>= fix.value,1:2]
KAA<-dm[dm[,3]>= minlevel,1:2]
## determination of eval.area:
areanew<-Areanew(Area, KAA)
######  ************************************************************  ######
## Considering matrix of contours of possible groups:
MAT<-groups.fix(dnew, fix.value)
## Matrix of Density:
## -) If K is empty:
if(nrow(matrix(K, ncol=2))==0) MM<-cbind(0,0,0)
## -) If K has only one row:
if(nrow(matrix(K, ncol=2))==1){
  K<-matrix(K, ncol=2)
```

```
38    RES<-point.in.polygon(MAT[MAT[,3]==1,1],MAT[MAT[,3]==1,2], areanew[,1],
39                          areanew[,2])
40    ## contour is strict outside of the eval.area:
41      if(all(RES==0)){
42         MAT[,1:3]<- 0
43         MM<-cbind(0,0,0)
44      }
45    ## contour is within or at the border of the eval.area:
46      if(any(RES)!=0){
47         MAT<-MAT
48    ## testing which density points are within the eval.area:
49        test<-point.in.polygon(K[,1], K[,2], areanew[,1] , areanew[,2])
50         r<-which(test>0, arr.ind=TRUE)
51         Knew<-K[r,]
52     ## Matrix of density with resp. to one grouping
53         MM<-cbind(matrix(Knew, ncol=2), 1)
54      }
55  }
56  ## -) K consist of more than one rows: **************************##################
57  if(nrow(matrix(K, ncol=2)) > 1){
58  ## case with only one group ---------------------- ###############
59    if(max(MAT[,3])==1){
60       RES<-point.in.polygon(MAT[MAT[,3]==1,1],MAT[MAT[,3]==1,2], areanew[,1],
61                             areanew[,2])
62    ## contour is strict outside of the eval.area:
63      if(all(RES==0)){
64         MAT[,1:3]<- 0
65         MM<-cbind(0,0,0)
66      }
67    ## contour is within or at the border of the eval.area:
68      if(any(RES)!=0){
69         MAT<-MAT
70     ## testing which density points are within the eval.area:
71       test<-point.in.polygon(K[,1], K[,2], areanew[,1] , areanew[,2])
72        r<-which(test>0, arr.ind=TRUE)
73        Knew<-K[r,]
74     ## Matrix of density with resp. to one grouping:
75        MM<-cbind(matrix(Knew, ncol=2), 1)
76
77      }
78    }
79   ############### ------------------------------- ##################
80
81  ## case with more than one group: ----------------------##########
82   if(max(MAT[,3])>1){
83    ## checking the contours of possible groups:
84     RES<-sapply(min(MAT[,3]):max(MAT[,3]), function(i){
85            point.in.polygon(MAT[MAT[,3]==i,1],MAT[MAT[,3]==i,2], areanew[,1],
86                             areanew[,2])})
87     for(i in min(MAT[,3]):max(MAT[,3])){
88        if(all(RES[[i]]==0)){
89   ## all contours outside of the eval.area are deleted:
90           MAT<-MAT[-which(MAT[,3]==i),]
91        }
92        else MAT<-MAT
```

```
93     }
94
95   if(nrow(MAT)==0) MM<-cbind(0,0,0)
96
97   if( nrow(MAT)>0 ){
98
99     if(min(MAT[,3])==max(MAT[,3])) {
100      test<-point.in.polygon(K[,1], K[,2], areanew[,1] , areanew[,2])
101      r<-which(test>0, arr.ind=TRUE)
102      Knew<-K[r,]
103  ## Matrix of density with resp. to one grouping:
104      MM<-cbind(matrix(Knew, ncol=2), min(MAT[,3]))
105    }
106## determation which contours are at the border which are in the eval.area and
107## splitting the Matrix of contours into the one within the area and the one at
108## the border of the area:
109   if(min(MAT[,3])!=max(MAT[,3])){
110## MATborder.... Matrix of contours at the border
111     MATborder<-matrix(NA, ncol=3)
112## MATin .... Matrix of contours within the area
113     MATin<-matrix(NA, ncol=3)
114
115## vector of the classification of contours within
116     con<-vector()
117##vector of the classification of contours at the border
118     con.out<-vector()
119
120     n<-1
121     k<-1
122
123    for(i in min(MAT[,3]): max(MAT[,3])){
124       if(all(RES[[i]]==1)){
125        con[n]<-i
126        n<-n+1
127       }
128       if(length(con)>=1){
129         rin<-sapply(1:length(con), function(i) nrow(MAT[MAT[,3]==con[i], ]))
130         crin<-cumsum(rin)
131         m<-1
132         MATin<-matrix(nrow=sum(rin), ncol=3)
133          for(j in 1:length(con)){
134                 MATin[m:crin[j], ]<-MAT[MAT[,3]==con[j],]
135                 m<-crin[j]+1
136          }
137       }
138      if(any(RES[[i]]==2) || any(RES[[i]]==3) ||
139        (any(RES[[i]])==1 & all(RES[[i]])!=1)){
140          con.out[k]<-i
141          k<-k+1
142      }
143      if(length(con.out)>=1){
144           r<-sapply(1:length(con.out),
145                       function(i) nrow(MAT[MAT[,3]==con.out[i],]))
146           cr<-cumsum(r)
147           m<-1
```

```
148            MATborder<-matrix(nrow=sum(r), ncol=3)
149             for(j in 1:length(con.out)){
150                  MATborder[m:cr[j],] <- MAT[MAT[,3]==con.out[j], ]
151                  m<-cr[j]+1
152             }
153       }
154     }
155
156 ## checking which density points are within the eval.area:
157     test<-point.in.polygon(K[,1], K[,2], areanew[,1],areanew[,2])
158     r<-which(test>0, arr.ind=TRUE)
159     Knew<-K[r,]
160 ## Knew is matrix of density points within the eval.area
161 ## Checking of contours within the eval.area:
162 ## a) case no contours within:
163    if(nrow(na.omit(MATin))==0) MMin<-matrix(NA, ncol=3)
164
165  if(nrow(na.omit(MATin))>0){
166     MMin<-matrix(NA, ncol=3)
167
168 ## b) only 1 contour within:
169    if((min(MATin[,3])==max(MATin[,3]))){
170       test1<-point.in.polygon(Knew[,1], Knew[,2],
171       MATin[MATin[,3]==min(MATin[,3]),1] , MATin[MATin[,3]==min(MATin[,3]),2])
172       rgr<- which(test1!=0, arr.ind=TRUE)
173
174       if(length(rgr)>0){
175          MMin<-cbind(matrix(Knew[rgr, ], ncol=2), min(MATin[,3]))
176       }
177       if(length(rgr)==0) MMin<-matrix(NA, ncol=3)
178    }
179 ## c) more than one contour within:
180    if(min(MATin[,3])!=max(MATin[,3])){
181       test1<-vector("list",length(min(MATin[,3]):max(MATin[,3])))
182       l<-vector()
183       for(i in min(MATin[,3]):max(MATin[,3]) ){
184          test1[[i]]<-point.in.polygon(Knew[,1], Knew[,2],
185                         MATin[MATin[,3]==i,1], MATin[MATin[,3]==i,2])
186       }
187      rg<-lapply(1:length(test1),function(i) which((test1[[i]])!=0,arr.ind=TRUE))
188      for(i in 1:length(rg)){
189         if(all(is.integer(0)==rg[[i]])==FALSE) l[i]<-i
190      }
191      l<-na.omit(l)
192 ## Converting the list into matrix:
193      if(is.logical(l)!=TRUE){
194       if(length(l)==1) MMin<-cbind(matrix(Knew[rg[[l]],], ncol=2),l)
195       if(length(l)>1){
196            KK<-lapply(1:length(l),
197                       function(i) cbind(matrix(Knew[rg[[l[i]]],], ncol=2),l[i]))
198            nr<-sapply(1:length(KK), function(x) nrow(KK[[x]]))
199            csnr<-cumsum(nr)
200            MMin<-matrix(nrow=(sum(nr)), ncol=3)
201            MMin[1:csnr[1], ] <- KK[[1]]
202         for(i in 2:length(nr)){
```

```
203                MMin[(csnr[i-1]+1):(csnr[i]),] <- KK[[i]]
204          }
205        }
206       }
207      }
208  }
209 ## Deleting the points from Knew, which have been already applied above to the
210 ## corresponding groups:
211   gin<-sapply(1:nrow(Knew), function(i) which((Knew[i,1]==MMin[,1] &
212                                          Knew[i,2]==MMin[,2]), arr.ind=TRUE))
213   indin<-vector()
214    for(i in 1:length(gin)){
215      if(length(gin[[i]])==0) indin[i]<-i
216    }
217   indin<-na.omit(indin)
218   Knew<-Knew[indin, ]
219
220 ## Checking the remaining density points,to which contour at border they belong
221 ## a) case if there is no contour at border:
222    if(nrow(na.omit(MATborder))==0) MMb<-matrix(NA, ncol=3)
223 ## SubMatrix of border contours:
224  MMbor<-matrix(NA, ncol=3)
225
226  if(nrow(na.omit(MATborder))>0){
227      MATb<-MATborder
228
229 ## b) case if only one contour at border:
230   if(length(con.out)==1){
231      pol<-Areanew(MATb[,1:2], KAA)
232      testAA<-point.in.polygon(Knew[,1], Knew[,2], pol[,1], pol[,2])
233      rr<-which(testAA>0, arr.ind=TRUE)
234 ##MMb Matrix of density with resp. contours
235      if(length(rr)>0){
236         MMb<- cbind(matrix(Knew[rr,], ncol=2), con.out)
237      }
238    }
239 ## c) case more than one contour at border:
240   if(length(con.out)>1){
241     kont<-sapply(1:length(con.out),
242                       function(x) nrow(MATb[MATb[,3]==con.out[x],]))
243     KA<-list()
244     Kn<-Knew
245     for(i in 1:length(con.out)){
246      if(nrow(Kn)>0){
247         nrkon<-which(kont==max(kont))
248
249        if(length(nrkon)>1){nrkon<-nrkon[1]}
250        if(length(nrkon)==1) nrkon<-nrkon
251
252        AA<-MATb[MATb[,3]==con.out[nrkon],]
253        ppoly<-Areanew(AA[,1:2], KAA)
254        testAA<-point.in.polygon(Kn[,1], Kn[,2], ppoly[,1], ppoly[,2] )
255        rr<-which(testAA>0, arr.ind=TRUE)
256        if(length(rr>0)){
257             KA[[i]]<- cbind(matrix(Kn[rr,], ncol=2), con.out[nrkon])
```

```
258         }
259         if(length(rr)==0)  KA[[i]]<- NA
260         if(any(!is.na(KA[[i]]))){
261            ind1<-vector()
262            gg<- sapply(1:nrow(Kn), function(x) which((Kn[x,1]==KA[[i]][,1])
263                                        & (Kn[x,2]==KA[[i]][,2]), arr.ind=TRUE))
264           for(j in 1:length(gg)){
265              if(length(gg[[j]])==0) ind1[j]<-j
266           }
267           ind1<-na.omit(ind1)
268           Kn<-matrix(Kn[ind1, ], ncol=2 )
269         }
270        MATb<-MATb[-which(MATb[,3]==con.out[nrkon]), ]
271        kont[nrkon]<-0
272       }
273      }
274    if(any(!is.na(KA))){
275      nrb<-vector()
276    ## Checking for NA's
277      for(x in 1:length(KA)){
278        if(all(!is.na(KA[[x]]))) nrb[x] <- nrow(KA[[x]])
279        else nrb[x] <- 0
280      }
281    ## Converting the list KA into Matrix
282       csnrb<-cumsum(nrb[nrb>0])
283       MMb<-matrix(nrow=sum(nrb), ncol=3)
284       if(any(nrb==0)){
285            KA<-KA[-which(nrb==0)]
286       }
287     if(all(nrb!=0)) KA<-KA
288     if(!is.list(KA)) MMb<-KA
289     if(is.list(KA)){
290         MMb[1:csnrb[1], ] <- KA[[1]]
291       if(length(csnrb)>1){
292          for(i in 2:length(csnrb)){
293             MMb[(csnrb[i-1]+1):(csnrb[i]),] <- KA[[i]]
294          }
295       }
296      }
297    }
298 ## Deleting the applied density points of Knew
299    g<-sapply(1:nrow(Knew), function(i)  which((Knew[i,1]==MMb[,1] &
300                                          Knew[i,2]==MMb[,2]), arr.ind=TRUE))
301    ind<-vector()
302    for(i in 1:length(g)){
303       if(length(g[[i]])==0) ind[i]<-i
304    }
305    ind<-na.omit(ind)
306    Knew<-Knew[ind, ]
307 ## case if there are not any density points left
308    if(nrow(na.omit(Knew))==0) MMb<-MMb
309 ## case checking of the remained density points, where they belong:
310    if(nrow(na.omit(Knew))>0){
311      testborder<-vector("list",length(min(MATborder[,3]):max(MATborder[,3])))
312      lborder<-vector()
```

```
    for(i in min(MATborder[,3]):max(MATborder[,3]) ){
     testborder[[i]]<-point.in.polygon(Knew[,1], Knew[,2],
            c(MATborder[MATborder[,3]==i,1]) , c(MATborder[MATborder[,3]==i,2]))
    }
    rborder<-sapply(1:length(testborder),function(i) which((testborder[[i]])!=0,
                                                    arr.ind=TRUE))
    for(i in 1:length(rborder)){
        if(all(is.integer(0)==rborder[[i]])==FALSE) lborder[i]<-i
    }
    lborder<-na.omit(lborder)
    if(length(lborder)==1){
        MMbor<- cbind(matrix(Knew[rborder[[lborder[1]]],], ncol=2),lborder[1])
    }
    if(length(lborder)>1){
        Kbor<-sapply(1:length(lborder), function(i)
                  cbind(matrix(Knew[rborder[[lborder[i]]],], ncol=2),lborder[i]))
        nrbor<-sapply(1:length(Kbor), function(x) nrow(Kbor[[x]]))
        csnrbor<-cumsum(nrbor)
        MMbor<-matrix(nrow=(sum(nrbor)), ncol=3)
        MMbor[1:csnrbor[1], ] <- Kbor[[1]]
      for(i in 2:length(nrbor)){
        MMbor[(csnrbor[i-1]+1):(csnrbor[i]),] <- Kbor[[i]]
      }
    }
  }
 }
}
MM<-na.omit(rbind(MMb, MMbor, MMin))
}
}
}
######     ----------------------- ######################
}
### *************************** ########################################
 return(list(Dpoints=MM, Area=areanew))
}
)
```

A.16 NGroups()

With this function the number of groupings within the eval.area of a sector per level cut, hence the third indicator, is calculated.

```
## Function for Indicator 3, computes the number of Groupings per level:
setGeneric("NGroups", function(Dp){ standardGeneric("NGroups") })
setMethod("NGroups", signature=c("matrix"), definition=function(Dp){
   con<-Dp[,3]
   if(all(con>0)){
   test<-sapply(1:length(con), function(x) con[x]==con[x+1])
   ind<-which(test==FALSE)
```

```
   contours<-(1+length(ind))
   }
   if(all(con==0))  contours<-0

 return(contours)
}
)
```

A.17 separation()

The function below enables the calculation of the last indicator *modCHI* for a sector per level cut.

```
## Function for the computing of the indicator Sep:
library(stats)
setGeneric("separation", function(MM){standardGeneric("separation")})
setMethod("separation", signature=c("matrix"), definition= function(MM){
## case if there is no density points > fix.value :
 if(all(MM==0)){
    IND <- NA
    kern<-NA
 }
## case for only one grouping:
 if(min(MM[,3])==max(MM[,3]) & any(MM > 0)){
 ## defining of center of grouping
  mx<- mean(c(MM[,1]))
  my<- mean(c(MM[,2]))
  kern<-cbind(na.omit(mx), na.omit(my))
  IND <- 0
 }
## case for more than one grouping:
 if((min(MM[,3])!= max(MM[,3]))){
   ind<-lapply(seq(min(MM[,3]),max(MM[,3]),1),
               function(i) if(length(MM[MM[,3]==i,])!=0){ which(MM[,3]==i)} )
   pp<-sapply(1:length(ind), function(i) (length(ind[[i]])>0))
   ind<-ind[pp==TRUE]
   ## center
   mx<-sapply(seq(1,length(ind),1), function(i) mean(c(MM[,1][ind[[i]]])))
   my<-sapply(seq(1,length(ind),1), function(i) mean(c(MM[,2][ind[[i]]])))
   kern<-cbind(na.omit(mx), na.omit(my))
## calculating WSA:
  WS<-sapply(seq(1,length(ind),1),
       function(j) sum(abs(MM[,1][ind[[j]]]-mx[j])+abs(MM[,2][ind[[j]]]-my[j])))
  WSS<-sum(na.omit(WS))
## calculating BSA:
  BS<-dist(kern, method="manhattan", diag=F)
  BSS<- sum(BS)
## INDIKATOR:
  IND1<-BSS/WSS
  IND<-IND1*(nrow(MM)-nrow(kern))/(nrow(kern)-1)
```

```
 }
return(list(INDIKATOR=IND, Centre=kern))
}
)
```

A.18 eval.sep()

This function is supposed to compute the third and fourth indicator for all sectors of a sample in one step.

```
## Function for evaluation of IND3 and IND4 in one step for all Sectors and
## for all level cuts:
setGeneric("eval.sep", function(dens, alpha, init, ndens){standardGeneric("eval.sep")})
setMethod("eval.sep", signature=c("list", "vector", "numeric", "numeric"),
definition= function(dens, alpha, init, ndens){

 dpoints<-list()
 Aread<-list()
 IND3<-matrix(nrow=length(alpha), ncol=(ndens-init+1))
 IND4<-matrix(nrow=length(alpha), ncol=(ndens-init+1))
 centre<-list()
 vec<-seq(init, ndens, 1)
 for(i in 1:(ndens-init+1)){
   d<-dens[[vec[i]]]
   dpoints[[i]]<- sapply(1:length(alpha),
                     function(x) grouping(d, alpha[1], alpha, alpha[x])$Dpoints)
   Aread[[i]]<-  grouping(d, alpha[1], alpha, alpha[1])$Area
## IND3: columns represent sectors, rows the level cuts
 IND3[,i]<-sapply(1:length(alpha), function(x) NGroups(dpoints[[i]][[x]]))
## IND4 sep:
 IND4[,i]<-sapply(1:length(alpha),
                    function(x) separation(dpoints[[i]][[x]])$INDIKATOR)
## center of groupings
 centre[[i]]<-sapply(1:length(alpha),
                    function(x) separation(dpoints[[i]][[x]])$Centre)
 }
return(list(Earea=Aread,Dp=dpoints, cent=centre, IND3=IND3, IND4=IND4))
}
)
```

A.19 Script

In the following the main steps for the evaluation are described, based on Sample 1:

```
### Evaluation :
rm(list=ls())
```

```
3
4  ############  ********************************************** #################
5  ## load data (previously imported via data.read())
6  load("daten.RData")
7  ## optimal bandwidth (previously computed via Bandwidth())
8  load("bw.RData")
9  ## density estimation of both Samples as list (previously computed via kde2())
10 load("dichten.RData")
11
12 ## computing minlevel:
13 minlevel<-baseline(dichten)
14 maxima<-sapply(1:length(dichten), function(i) max(dichten[[i]]$z))
15 upperlimit<-max(maxima)
16 ## level cuts alpha:
17 alpha<-seq(minlevel, upperlimit, length.out=50)
18
19 ## computing bivariate uniform distribution:
20 x.vek<-c(0,100)
21 y.vek<-c(0,100)
22 tgvt<-gvt2d(x.vek,y.vek, 100)
23 maxtgvt<-max(tgvt$z)
24
25 ## Evaluation by the example of Sample 1,
26 ## with sectors of density: dichten[[1]] to dichten[[14]]:
27
28 ####### ************************************************************** #######
29 # 1) Indikator giniUTR:   Comparison with ~U:
30 ## scale tgvt, so that max(tgvt)=upperlimit
31 fak<-upperlimit/maxtgvt
32 newtgvtt<-list(x=tgvt$x, y=tgvt$y, z=tgvt$z*fak)
33 UTRg<-UTR(newtgvtt, minlevel, alpha)
34
35 UTRd<-Inequ(dichten,alpha,1,14)$UTR
36
37 IND1<-Inequ(dichten,alpha,1,14)$Inequ
38
39 ### Plot UTR Sample1:
40 vcol<-palette(gray(seq(0,0.9,len=14)) )
41 plot(alpha,UTRd[,1], type="l", col=vcol[2] ,lwd=2, xlab="Level Cuts", ylab="UTR")
42 lines(alpha,UTRd[,2], lty=5, lwd=2, col=vcol[3])
43 lines(alpha,UTRd[,3], lty=2, lwd=2, col=vcol[4])
44 lines(alpha,UTRd[,4], lty=4, lwd=2.5, col=vcol[8])
45 lines(alpha,UTRd[,5], lty=6, lwd=2, col=vcol[6])
46 ## tgvt:
47 lines(alpha,UTRg, lwd=2, col=vcol[9])
48 title(main="UTR ")
49 legend("right",  legend=c("S_1", "S_2", "S_3", "S_4","S_5", "~U"),
50  col=c(vcol[2],vcol[3],vcol[4],vcol[8],vcol[6],vcol[9]), lty=c(1,5,2,4,6,1),
51   lwd=c(2,2,2,2.5,2,2))
52
53 ### Log-Plot UTR Sample1:
54 plot(log="x",UTRd[,1], type="l", col=vcol[2] ,lwd=2, xlab="Log-Level Cuts", ylab="UTR")
55 lines(UTRd[,2], lwd=2, col=vcol[3] ,lty=2)
56 lines(UTRd[,3], lwd=2, lty=3, col=vcol[4] )
57 lines(UTRd[,4], lwd=2.5, lty=4, col=vcol[8] )
```

```
lines(UTRd[,5], lwd=2, lty=5, col=vcol[6] )
## tgvt:
lines(UTRg, col=vcol[9], lwd=2, lty=1)
title(main="UTR on a logarithmic Scale")
legend("right", legend=c("S_1", "S_2", "S_3", "S_4","S_5", "~U"),
 lty=c(1,5,2,4,6,1),col=c(vcol[2],vcol[3],vcol[4],vcol[8],vcol[6],vcol[9]),
 lwd=c(2,2,2,2.5,2,2))
## do this plots for all 14 sectors

### Plot Indikator 1 Gini:
plot(IND1, type="l", lwd=2, xlab="Sectors", ylab="giniUTR")
title(main="Gini-Index for all Sectors of Sample 1")
# rcol<-palette(gray(seq(0,0.9,len=14)) )
# rcol<-palette(rainbow(14))
rcol<-palette( gray(seq(0,0.9,len=15)))
barplot(IND1[,1], names.arg=c(1:14), col=rcol[1:14],
  ylim=c(0.0,1.0), xlab="Sectors", ylab="giniUTR")

# 2.) Comparison with biv. normal distr. :
x1<-seq(-10,10,length=100)
x2<-x1
tnvt<-nvt2d(x1,x2, s11=10, s22=10, rho=0)
maxtnvt<-max(tnvt$z)
## Scaling of data within the UTR()
## factor for ~U, so that: max(tgvt)=max(tnvt)
fak1<-maxtnvt/maxtgvt
newtgvt<-list(x=tgvt$x, y=tgvt$y, z=tgvt$z*fak1)

## adjusted level cuts:
nalpha<-seq(minlevel,maxtnvt, length.out=50)

utr.g<-UTR(newtgvt, minlevel, nalpha)

## nUTR analytically computed:
n.utr<-nUTR(10, nalpha)
## Scaled UTR's of Sectors 1:14  :
UTRd<-CompNdist(dichten,nalpha,1,14,10,maxtnvt)$sUTR
## Indicator 2 for Sample 1:
IND2<-CompNdist(dichten,nalpha,1,14,10,maxtnvt)$CNdist

rcol<-palette( gray(seq(0,0.9,len=15)))
### Plot scaled UTR Sample1 vs nUTR for all Sectors:
plot(nalpha,n.utr, type="l", col=rcol[1], lwd=2.5, xlab="Level Cuts", ylab="UTR")
lines(nalpha,UTRd[,1], lwd=1.5,col=rcol[2], lty=1)
lines(nalpha,UTRd[,2], lwd=2,col=rcol[3], lty=2)
lines(nalpha,UTRd[,3], lwd=2,col=rcol[4], lty=3)
lines(nalpha,UTRd[,4], lwd=2,col=rcol[5], lty=4)
lines(nalpha,UTRd[,5], lwd=2,col=rcol[6], lty=5)
## tgvt:
lines(nalpha,utr.g, lwd=2, col=rcol[8], lty=1)
title(main="Scaled UTRs vs nUTR")
legend("right", legend=c("nUTR","S_1","S_2","S_3","S_4","S_5",
              "~U"), col=c(rcol[1:6], rcol[8]),
 lty=c(1,1,2,3,4,5,1), lwd=c(2.5,1.5,2,2,2,2,2))

```

```
### Plot Indikator 2 cpUTR for Sample 1:
plot(IND2, type="l", lwd=2, xlab="Sectors", ylab="cpUTR")
title(main="Indicator cpUTR for all Sectors of Sample 1")
barplot(IND2[,1],names.arg=c(1:14),  col=rcol[1:14], ylim=c(0.0,0.15),
        xlab="Sectors", ylab="cpUTR")


######## ********************************************************** ########
## 3.) Computing NGroups and modCHI:
alpha<-seq(minlevel, upperlimit, length.out=50)
#result<-eval.sep(dichten, alpha, 1, 14)
#save(file="result1250.RData", "result")
load("result1250.RData")

## IND 3: NGroups forall Sectors:
IND3<- result$IND3
## IND 4: Separation (modCHI) forall Sectors:
IND4<-result$IND4
## do the plots forall sectors!
rcol<-palette(rainbow(14))
plot(alpha, IND3[,11], type="l", col=rcol[1],lwd=1,xlab="Level Cuts",ylab="NGroups",ylim=c
    (0,10))
lines(alpha, IND3[,12], lwd=2, col=rcol[2], lty=2)
lines(alpha, IND3[,13], lwd=2,col=rcol[3], lty=3)
lines(alpha, IND3[,14], lwd=2, col=rcol[4], lty=4)
lines(alpha, IND3[,10], lwd=2, col=rcol[5], lty=5)
title(main="Indicator NGroups for Sample 1")
legend("topright", legend=c("S_1", "S_2","S_3","S_4", "S_5"),
 lty=c(1,2,3,4,5), col=c(rcol[1:5]), lwd=c(1,2,2,2,2))

## plot IND 4 Sep:
plot(alpha, IND4[,1], type="l", col=rcol[1], lwd=1,xlab="Level Cuts",ylab="modCHI",ylim=c
    (0,40))
lines(alpha, IND4[,2], lwd=2,col=rcol[2], lty=2)
lines(alpha, IND4[,3], lwd=2,col=rcol[3], lty=6)
lines(alpha, IND4[,4], lwd=2, col=rcol[4], lty=4)
lines(alpha, IND4[,5], lwd=2,col=rcol[5], lty=5)
title(main="Indicator modCHI for Sample 1")
legend("topright", legend=c("S_1", "S_2", "S_3","S_4", "S_5"),
 lty=c(1,2,6,4,5),col=c(rcol[1:5]), lwd=c(1,2,2,2,2))

## Dp... Density points:
Dp<-result$Dp
## cent... centers of the groupings:
cent<-result$cent
## eval.area:
eval.area<-result$Earea

## plot eval.area at level with maximal groupings in the sectors:
mxngr<-sapply(1:ncol(IND3), function(i) which(IND3[,i]==max(IND3[,i])) )
## ergibt eine liste 1:14
mngr<-sapply(1:length(mxngr), function(i) mxngr[[i]][1])
## mngr is vector, where Ngroups is max.
## if max not unique, select the first level where Ngroups is max

```

```
166 for(i in 1:14){
167 x11()
168 plot(eval.area[[i]], type="l", lwd=2, xlim=c(0,5000), ylim=c(0,5000),
169      xlab="x-coord", ylab="y-coord")
170 title(main="Grouping within Density Estimation", line=2.3)
171 mtext(bquote(Level == .(alpha[mngr[i]])), side=3, line=0.8, cex=0.7)
172 points(result$Dp[[i]][[mngr[i]]],col=result$Dp[[i]][[mngr[i]]][,3],pch=19,
173        cex=0.5)
174 points(result$cent[[i]][[mngr[i]]], pch="*", cex=1.5,col=1)
175 }
176
177 ####### ********************************************************** #########
```

Bibliography

G. Arminger. Mehrdimensionale stetige Verteilungen. Online aus Skriptum für die Vorlesungen Statistik I und II, 2009. Available online at: http://www.statistik.uni-wuppertal.de/fileadmin/dateien/lehre/grundstudium/statistik_I/r_sektion/5.2-MehrdimensionalestetigeVerteilungen(PraktischerTeil).pdf.

American Society of Clinical Oncology ASCO. Brain Tumor. Website, 2012. Available online at: http://www.cancer.net/patient/Cancer+Types/Brain+Tumor.

Statistik Austria. Website, Oktober 2011a. Available online at: http://www.statistik.at/web_de/statistiken/gesundheit/krebserkrankungen/gehirn_zentralnervensystem/index.html.

Statistik Austria. Website, Oktober 2011b. Available online at: http://www.statistik.at/web_de/statistiken/gesundheit/krebserkrankungen/malignome_insgesamt/index.html.

T.C. Bailey and A.C. Gatrell. *Interactive Spatial Data Analysis*. Longman Scientific and Technical, England, 1995.

J. Bertolini. Tumorpathologie. Online, 2012. Available as pdf-file online at: http://pathologie.uniklinikum-leipzig.de/deutsch/download/lehre/ss12/JuliaBertolini/Tumorpathologie.pdf.

C.S. Bjornsson, G. Lin, Y. Al-Kofahi, A. Narayanaswamy, L.K. Smith, W. Shain, and B. Roysam. Associative Image Analysis: A Method for Automated Quantification of 3D multi-paramater Images of Brain Tissue. *Journal of Neuroscience Methods*, 170:165–178, 2008.

J.N. Bruce. Ependymoma. Website, January 2009. Available online at: http://emedicine.medscape.com/article/277621-overview.

R. Calinski and J. Harabasz. A Dendrite Methode For Cluster Analysis. *Communications in Statistics*, 3:1–27, 1974.

Foundation CERN. Pediatric Ependymoma Images. Website, 2011a. Available online at: `http://www.cern-foundation.org/Content.aspx?id=608`.

Foundation CERN. MRI. Website, 2011b. Available online at: `http://www.cern-foundation.org/Content.aspx?id=588`.

E. Cramer and U. Kamps. *Grundlagen der Wahrscheinlichkeitsrechnung und Statistik*. Springer Verlag, Berlin, 2008.

G. Faes. Website, September 2007. Available online at: `http://www.faes.de/Basis/Basis-Lexikon/Basis-Lexikon-Lorenz-Kurve/basis-lexikon-lorenz-kurve.html`.

R. Francois. R Graph Gallery. Website, September 2011a. Available online at: `http://addictedtor.free.fr/graphiques/sources/source_42.R`.

R. Francois. R Graph Gallery. Website, September 2011b. Available online at: `http://addictedtor.free.fr/graphiques/sources/source_1.R`.

R.C. Gonzales and R.E. Woods. *Digital Image Processing*. Pearson/Prentice Hall, Upper Saddle River, New Jersey, 2008.

B. Grala, T. Markiewicz, W. Kozlowski, S. Osowski, J. Slodkowska, and W. Papierz. New Automated Image Analysis Method for the Assessment of Ki-67 labeling Index in Meningiomas. *Folia Histochemica Et Cytobiologica*, 47:587–592, 2009.

J. Groß. *A Normal Distribution Course*. Peter Lang Verlag, Frankfurt am Main, 2004.

Hamamatsu. Virtual microscopy / nanozoomer. Website, March 2012. Available online at: `http://sales.hamamatsu.com/en/products/system-division/virtual-microscopy.php`.

W. Härdle, M. Müller, S. Sperlich, and A. Werwatz. *Nonparametric and Semiparametric Models*. Springer-Verlag, Berlin New York, 2004.

T. Hermes. *Digitale Bildverarbeitung*. Carl Hanser Verlag, München Wien, 2005.

H. Kütting and M.J. Sauer. *Elementare Stochastik*. Spektrum Akademischer Verlag, Imprint von Springer, Heidelberg, 2011.

F. Li, F.S. Moiseiwitsch, and V.R. Korostyshevskiy. Region-Based Statistical Analysis of 2D PAGE Images. *Computational Statistics & Data Analysis*, 55:3059–3072, 2011.

U. Ligges. *Programmieren mit R.* Springer Verlag, Dortmund, 2008.

U. Maulik and S. Bandyopadhyay. Performance Evaluation of Some Clustering Algorithms and Validity Indices. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24:1650–1654, 2002.

National Institute of Neurological Disorders & Stroke NINDS. Brain and Spinal Tumors: Hope Through Research. Website, August 2011. Available online at: `http://www.ninds.nih.gov/disorders/brainandspinaltumors/detail_brainandspinaltumors.htm#182773060`.

N. Otsu. A Threshold Selection Method from Gray-Level Histograms. *IEEE Transactions on Systems, man and Cybernetics*, 9(1):62–66, January 1979.

M. Preusser, H. Heinzl, E. Gelpi, R. Höftberger, I. Fischer, I. Pipp, I. Milenkovic, A. Wöhrer, F. Popovici, S. Wolfsberger, and J.A. Hainfellner. Ki67 index in intracranial ependymoma: a promising histopathological candidate biomarker. *Histopathology*, 53: 39–47, 2008.

R. Open-Source Software. Website, 2011. `http://www.r-project.org/`.

L. Sachs and J. Hedderich. *Angewandte Statistik, Methodensammlung mit R.* Springer, Berlin, 2006.

R. Schlittgen. *Multivariate Statistik.* Oldenbourg Wissenschaftsverlag München, München, 2009.

D.W. Scott. *Multivariate Density Estimation.* Wiley-Interscience, John Wiley & Sons INC., New York, Chichester, Brisbane, Toronto, Singapure, 1992.

S.J. Sheather and M.C. Jones. A reliable data-based bandwidth selection method for kernel density estimation. *Journal of the Royal Statistical Society series B*, 53:683–690, 1991.

B.W. Silverman. *Density Estimation.* Chapman and Hall, London, 1986.

T.T. Soong. *Fundamentals of Probability and Statistics for Engineers.* John Wiley and Sons Ltd , The Atrium, Southern Gate, Chichester, West Sussex PO19 8SQ, England , 2004.

H. Travis. University of texas inequality project: The theoretical basics of popular inequality measures. Online, January 2008. Available online at: `http://utip.gov.utexas.edu/tutorials/`.

Cancer Research UK. Brain Tumours. Website, 2012. Available online at: `http://cancerhelp.cancerresearchuk.org/type/brain-tumour`.

W.N. Venables and B.D. Ripley. *Modern Applied Statistics with S-Plus*. Springer, New York and Berlin and Heidelberg, 1994.

C. Wählby, I.-M. Sintorn, F. Erlandsson, G. Borgefors, and E. Bengtsson. Combining Intensity, Edge and Shape Information for 2D and 3D Segmentation of cell nuclei in tissue sections. *Journal of Microscopy*, 215:67–76, July 2004.

A. Walser. Automatisierte Auswertung der Zellproliferation in menschlichen Gehirntumoren. Master's thesis, Vienna University of Technology, 2011.