NTNU
Norwegian University of
Science and Technology

Exact Statistical Inference in Parametric Models

Methods for Constructing Confidence Intervals and Confidence Regions Based on Conditional Parametric Bootstrap and Data Depth

Audun Sektnan

Master of Science in Physics and Mathematics
Submission date: June 2017
Supervisor: Bo Henry Lindqvist, IMF

Norwegian University of Science and Technology
Department of Mathematical Sciences

Problem Description

- Study methods for generating approximate and exact confidence intervals and confidence regions for parametric models with one or more unknown parameters, using:
  - Conditional parametric bootstrapping
  - Methods based on data depth
- Study in particular the construction of confidence intervals and confidence regions for the gamma distribution

Assignment given: January 15, 2017

Supervisor: Bo H. Lindqvist

Preface

This thesis is written as part of my Master of Science degree at the Norwegian University of Science and Technology (NTNU). The work was done in the spring of 2017 at the Department of Mathematical Sciences.

I would like to thank my supervisor professor Bo Henry Lindqvist for helping me in my work, and always being available to answer all my questions.

Audun Sektnan

Trondheim, June 2017

Abstract

In this Master's thesis we investigate approaches for constructing approximate and exact confidence intervals and regions in parametric models. A supposedly exact method, called conditional parametric bootstrap, is used to generate confidence intervals for the parameters in the gamma distribution. However, simulation studies are carried out that question the correctness of this method. More precisely, the scale parameter seems to obtain a higher coverage probability than expected. The results are compared to approximate intervals using the more familiar bootstrap methods.

Next, we look at a concept called data depth, and apply it on two-dimensional distributions and data sets. This can be used to order multidimensional data, and here we analyze some of the well known types of depths. These different types are then used, in combination with methods from the conditional parametric bootstrap, to construct approximate confidence regions for the parameters in the gamma distribution. The coverage probabilities are analyzed, and we observe how one can obtain close to exact confidence regions just by adjusting a simulation parameter in the algorithm.

Sammendrag

I denne masteroppgaven undersøker vi måter å konstruere tilnærmede og eksakte konfidensintervall og konfidensområder i parametriske modeller. En antatt eksakt metode, kalt ”conditional parametric bootstrap”, brukes til å generere konfidensintervaller for parametrene i gammafordelingen. Simuleringsstudier gjennomføres som derimot setter spørsmålstegn ved hvor korrekt denne metoden er. Mer presist så ser det ut til at skalaparameteren oppnår en for høy andel av simuleringene innenfor konfidensintervallet, ut fra det som er forventet verdi. Resultatene sammenlignes med tilnærmede intervaller funnet ved bruk at de mer kjente bootstrapmetodene.

Deretter ser vi på et konsept kalt datadybde, og anvender det på todimensjonale fordelinger og datasett. Dette kan brukes til å ordne multidimensjonale data, og her analyserer vi noen av de mer kjente dybdetypene. Disse ulike dybdene brukes så, i kombinasjon med metoder fra ”conditional parametric bootstrap”, til å konstruere tilnærmede konfidensområder for parametrene i gammafordelingen. Andelen av simuleringene som faller innenfor konfidensområdet analyseres, og vi observerer at en kan oppnå et nesten eksakt konfidensområde bare ved å justere på en simuleringsparameter i algoritmen.

Contents

1 Introduction **1**

2 Theory **3**

2.1 Gamma Distribution . . . 3

2.2 Exact Confidence Intervals . . . 4

2.3 Bootstrap Confidence Intervals . . . 6

2.4 Sufficiency . . . 6

3 Conditional Parametric Bootstrapping **9**

3.1 Bivariate Normal Distribution . . . 9

3.2 Gamma Distribution . . . 15

3.3 Results and Discussion . . . 17

4 Probability Regions **23**

4.1 Order Statistics - One Dimension . . . 24

4.2 Order Statistics - Two Dimensions . . . 26

4.3 Depth Statistics . . . 28

4.3.1 Mahalanobis Depth . . . 29

4.3.2 Simplicial Depth . . . 30

4.3.3 Adjusting the Simplicial Depth . . . 31

4.3.4 Tukey's Depth . . . 36

4.3.5 Angle Depth . . . 37

4.4 Probability Region from Depth . . . 39

4.4.1 Mahalanobis Depth . . . 41
4.4.2 Simplicial and Adjusted Simplicial Depth . . . 41
4.4.3 Convex Hull . . . 42
4.4.4 Tukey's Depth . . . 43
4.4.5 Angle Depth . . . 43
4.5 Plots of Probability Regions . . . 44
4.6 Coverage of Probability Regions from Depth Statistics . . . 48

5 Confidence Regions for the Gamma Distribution **53**
5.1 Results . . . 55
5.2 Logarithmic Transformation . . . 57
5.3 Coverage of Confidence Regions . . . 61
5.4 Conclusion . . . 64

Bibliography **66**

Chapter 1

Introduction

In statistics, parametric models denote families of probability distributions that are characterized by one or more model parameters. When doing statistical inference, one often assumes a particular model and tries to estimate the corresponding parameters using some data sample. This only makes sense if the model assumption is approximately correct, which often can be tested to some degree by various methods. Under the assumption of a particular parametric model, one has that there exists some true, but unknown, values for the model parameters. To assess the accuracy of an estimated parameter value, it is common to construct a confidence interval, which is a range of values that has a certain probability of containing the true parameter value.

The gamma distribution is a general and well known family of continuous distributions. It has, among others, both the exponential distribution and the chi squared distribution as special cases, and can take a variety of different shapes depending on the values of the two parameters that characterize the distribution. One common application is in the analysis of waiting times, following from the fact that waiting times between events in a Poisson process actually are gamma distributed. It has also been used in the prediction of rainfall, for instance in (Husak et al., 2007).

The distribution derives its name from the gamma function that appears in the normalization constant of the probability density function (see definitions and details in Section 2.1). It turns

out that this function, which is defined in terms of an integral, makes it somewhat tricky to compute the maximum likelihood estimators for the parameters, and also to construct exact confidence intervals.

A quite general method for constructing exact confidence intervals is introduced in (Lillegard and Engen, 1999). There they denote it as "conditional parametric bootstrapping", and this method is applicable on many parametric models. In this thesis we start by going over some statistical theory in Chapter 2 that we will use later, before we in Chapter 3 investigate how the conditional parametric bootstrap method works. At the end of this chapter we apply it on the gamma distribution, and study the correctness of the algorithm. Next, we will take a different approach to the inference in parametric models with two parameters, namely to look at confidence regions in $\mathbb{R}^2$ constructed using a concept called data depth. Various types of such depths are introduced in Chapter 4, where we also study how one can use this to generate what we will denote as "probability regions". The shapes and correctness are investigated at the end of the chapter, before we in Chapter 5 use a combination of this method and conditional parametric bootstrap to construct various confidence regions for the two parameters in the gamma distribution.

All the simulations and programming are done in R.

Chapter 2

Theory

Here we present some theory we will make use of later.

2.1 Gamma Distribution

The density of a gamma distributed variable is

$$f_X(x) = \frac{1}{\Gamma(\alpha)\beta^\alpha} x^{\alpha-1} \mathrm{e}^{-\frac{x}{\beta}}, \qquad 0 < x < \infty, \qquad \alpha > 0, \qquad \beta > 0, \tag{2.1}$$

where α and β are the shape and scale parameters, respectively (see for instance (Casella and Berger, 2002), Chapter 3, page 99). Here $\Gamma(\alpha)$ is the gamma function, defined as

$$\Gamma(\alpha) = \int_0^\infty x^{\alpha-1} e^{-x} dx.$$

Assuming a random sample $\mathbf{x} = (x_1, \ldots, x_n)$, the likelihood function can be written as

$$L(\alpha, \beta) = \left(\frac{1}{\Gamma(\alpha)\beta^\alpha}\right)^n \left(\prod_{i=1}^n x_i\right)^{\alpha-1} \mathrm{e}^{-\frac{1}{\beta}\sum_{i=1}^n x_i}. \tag{2.2}$$

The maximum likelihood estimates for α and β are found by finding the maximum of this function, or alternatively the log-likelihood function. The solutions cannot be written explicitly, but

can be found numerically, using for instance the function fitdistr from the library MASS in R. The equations to solve are the following:

$$\ln(\alpha) - \psi_0(\alpha) + c = 0,$$
$$\beta = \frac{1}{n\alpha}\sum_{i=1}^{n} x_i,$$

where $\psi_0(\alpha)$ is the digamma function,

$$\psi_0(\alpha) = \frac{d}{d\alpha}\ln(\Gamma(\alpha)),$$

and

$$c = \ln\left(\frac{\left(\prod_{i=1}^{n} x_i\right)^{1/n}}{\frac{1}{n}\sum_{i=1}^{n} x_i}\right).$$

2.2 Exact Confidence Intervals

Consider a random sample $\mathbf{X} = (X_1, X_2, \ldots, X_n)$, where each sample point is drawn from a known probability distribution $f_X(x;\theta)$ with an unknown one-dimensional parameter θ. Often the goal is to estimate the parameter θ by calculating some statistic $\hat{\theta}$ from the random sample. The next step could be to calculate an interval $[a, b]$ where you are quite certain (confident) that the true parameter θ lies, and this is called a *confidence interval*. The *coverage probability* is the probability that the real value θ is inside the confidence interval (see (Casella and Berger, 2002), Chapter 9, page 418). The *nominal* coverage probability (see (Hall, 1992), Chapter 1, page 12) is the desired probability for the real parameter θ to be inside the interval $[a, b]$, and often used values for this are 0.90, 0.95 or 0.99. In an *exact* confidence interval we have that the nominal and true coverage probabilities are equal.

Such exact confidence intervals are usually difficult to obtain for complex distributions, but can be found for some of the simpler ones. One general approach to finding such intervals is to find a variable, called a *pivot*, that is a function of both the parameter θ and the random sample $\mathbf{X}$ with a known distribution independent of θ. In the case of the univariate normal distribution, $X \sim$

$N(\mu,\sigma^2)$, there exists exact confidence intervals for both μ (with σ^2 either known or unknown) and σ^2 (with μ either known or unknown), see for instance (Ross, 2009), Chapter 7.3, page 242-243, 248 and 253-254. The case for σ^2 when μ is known is very similar to when μ is unknown, only replace $n-1$ with n in the chi-squared quantile, and use μ instead of $\bar{\mathbf{X}}$ in the calculation of S^2.

It is also possible to calculate an exact confidence interval for the parameter λ in the exponential distribution, by noting that

$$2\lambda \sum_{i=1}^{n} X_i \sim \chi^2_{2n}.$$

(see for instance (Ross, 2009), Chapter 7.6, page 267). Hence, a $(1-\alpha)100\%$ exact confidence interval is

$$\left[\frac{\chi^2_{2n,1-\frac{\alpha}{2}}}{2\sum_{i=1}^{n} x_i}, \frac{\chi^2_{2n,\frac{\alpha}{2}}}{2\sum_{i=1}^{n} x_i} \right],$$

where $\chi_{2n,\frac{\alpha}{2}}$ and $\chi_{2n,1-\frac{\alpha}{2}}$ are quantiles in the chi-squared distribution with $2n$ degrees of freedom.

Such exact confidence intervals are generally difficult to find, and so one often use some method to generate confidence intervals with coverage probability approximately equal to the nominal coverage probability.

Estimators can be evaluated using two criterias

- Is the estimator biased or unbiased?
- How small is the variance?

In a similar way one can evaluate a confidence interval by

- Is the true coverage probability equal to the nominal coverage probability?
- How small is the length of the confidence interval?

2.3 Bootstrap Confidence Intervals

Bootstrapping is a relatively easy way to make approximate confidence intervals when the distribution F is unknown, and was introduced in (Efron, 1979). The idea is to draw samples from some distribution F^* that approximates the true distribution F, and use many such resamples, or bootstrap samples, to make inference on the distribution of some statistic. A popular choice for the approximated distribution is the empirical distribution. Assuming a random sample $\mathbf{x} = (x_1, \ldots, x_n)$, the empirical distribution F^* puts a uniform probability $\frac{1}{n}$ on each of the sample points. This is what is called "the bootstrap method for the one-sample problem" in the paper by Efron. The empirical distribution will be a poor approximation of the true distribution F if n is small, but will get better as n increases, and will converge to the correct distribution F when n goes to infinity.

If the statistic is an estimator for some distribution parameter, one can use many such bootstrap samples to calculate an approximate confidence interval for the parameter. This can be done by calculating the value $T(\mathbf{x}^*)$ of the statistic using many samples $\mathbf{x}^* = \left(x_1^*, \ldots, x_n^*\right)$ drawn from the empirical distribution F^*, sorting these values, and then discarding $\frac{\alpha}{2}$ of the values in either end. One then ends up with a two-sided approximate $(1-\alpha)\%$ confidence interval, which is a type of *non-parametric* bootstrap confidence interval, because the resampling is done from the non-parametric empirical distribution.

Similarly, one can construct a *parametric* bootstrap confidence interval by resampling from the actual distribution F with parameters estimated from the random sample $\mathbf{x} = (x_1, \ldots, x_n)$. In this case one must assume that the data comes from a known (parametric) distribution F. For instance, assuming that F is the gamma distribution, one estimates the parameters $\hat{\alpha}$ and $\hat{\beta}$ from the data, and use these values to generate new bootstrap samples from Gamma$\left(\hat{\alpha}, \hat{\beta}\right)$.

2.4 Sufficiency

Consider a random sample $\mathbf{X} = (X_1, X_2, \ldots, X_n)$ where each X_i is from a known probability distribution $f_X(x;\theta)$ with an unknown parameter θ, possibly multidimensional. If the goal is to

estimate θ or a function $g(\theta)$, one would calculate some statistic $T(\mathbf{x})$ as an estimator, using the realization $\mathbf{x} = (x_1, x_2, \ldots, x_n)$. These estimates might be equal for different realizations $\mathbf{x}$ and $\mathbf{y}$, and one might wonder if it is possible to summarize the data from the sample $\mathbf{x}$ in such a way that no useful data is lost. This is the concept of sufficient statistics, which is defined as follows (see (Casella and Berger, 2002), Chapter 6, page 272).

A statistic $T(\mathbf{X})$ is a *sufficient statistic* for θ if the conditional distribution of the sample $\mathbf{X}$ given the value of $T(\mathbf{X})$ does not depend on θ.

A useful way of deciding if a particular statistic is indeed a sufficient statistic, is the Factorization Theorem (see (Casella and Berger, 2002), Chapter 6, page 276):

Factorization Theorem: Let $f(\mathbf{x};\theta)$ denote the joint pdf or pmf of a sample $\mathbf{X}$. A statistic $T(\mathbf{X})$ is a sufficient statistic for θ if and only if there exist functions $g(t;\theta)$ and $h(\mathbf{x})$ such that, for all sample points $\mathbf{x}$ and all parameter points θ,

$$f(\mathbf{x};\theta) = g(T(\mathbf{x});\theta)h(\mathbf{x}).$$

Chapter 3

Conditional Parametric Bootstrapping

A method for generating exact confidence intervals is described in (Lillegard and Engen, 1999), where they denote it as conditional parametric bootstrapping. This method works under quite general conditions in the case of models with only one parameter, and also for models with nuisance parameters. Confidence intervals are calculated for the parameters α and β in the gamma distribution using simulations. Before doing this, we look at a simpler case example to illustrate the method.

3.1 Bivariate Normal Distribution

The first example that is studied in (Lillegard and Engen, 1999) is the generation of an exact confidence interval for the correlation coefficient ρ in the bivariate normal distribution. Here we copy the approach, and the method works as follows. Let $(\mathbf{X},\mathbf{Y}) = \big((X_1,Y_1),\ldots,(X_n,Y_n)\big)$ be a random sample from the bivariate normal distribution, with the usual parameters μ_x, μ_y, σ_x, σ_y and ρ. Such a random sample can be generated using samples of univariate and independent random variables, $\mathbf{U} = (U_1,\ldots,U_n)$ and $\mathbf{V} = (V_1,\ldots,V_n)$, from the standard normal distribution,

$U_i, V_i \sim N(0,1)$, using the following formulas

$$X_i = \mu_x + \sigma_x U_i,$$
$$Y_i = \mu_y + \sigma_y \left[\rho U_i + (1-\rho^2)^{1/2} V_i\right],$$

for $i = 1,\ldots,n$. Next, each (X_i, Y_i) is transformed to the dependent random variables (Z_i, W_i) to get a distribution depending only on the single parameter ρ, using the following equations:

$$Z_i = \frac{X_i - \bar{X}}{S_x} = \frac{U_i - \bar{U}}{S_u}, \tag{3.1}$$
$$W_i = \frac{Y_i - \bar{Y}}{S_y} = \frac{\rho(U_i - \bar{U}) + (1-\rho^2)^{1/2}(V_i - \bar{V})}{\left[\rho^2 S_u^2 + (1-\rho^2) S_v^2 + 2\rho(1-\rho^2)^{1/2} r_{uv} S_u S_v\right]^2}, \tag{3.2}$$

for $i = 1,\ldots,n$. Here S_x, S_y, S_u, S_v and r_{uv} denotes the sample standard deviations and the sample correlation.

The maximum-likelihood estimate of ρ is

$$\hat{\rho} = \frac{1}{n}\sum_{i=1}^{n} Z_i W_i. \tag{3.3}$$

The main idea of the whole method is that, given a set of data values $(\mathbf{x}, \mathbf{y}) = (x_1, y_1), \ldots, (x_n, y_n)$, we can think of these data as being constructed from values for $(\mathbf{u}, \mathbf{v}) = (u_1, v_1), \ldots, (u_n, v_n)$ that we do not know. We can, however, generate new values $\mathbf{u}^*$ and $\mathbf{v}^*$ that are from the same distribution as the true values $\mathbf{u}$ and $\mathbf{v}$. The method is to keep the maximum-likelihood estimator $\hat{\rho}$ fixed, and to calculate what value of the parameter ρ that would give the same value for the estimate $\hat{\rho}$, using the new values of $\mathbf{u}^*$ and $\mathbf{v}^*$. Denoting this value $\tilde{\rho}$, this results in solving $\hat{\rho} = g(\tilde{\rho}; \mathbf{u}^*, \mathbf{v}^*)$ for $\tilde{\rho}$, where the function $g(\cdot)$ is defined as

$$g(\rho; \mathbf{U}, \mathbf{V}) = \frac{1}{n}\sum_{i=1}^{n}\left[\left(\frac{U_i - \bar{U}}{S_u}\right)\left(\frac{\rho(U_i - \bar{U}) + (1-\rho^2)^{1/2}(V_i - \bar{V})}{\left[\rho^2 S_u^2 + (1-\rho^2) S_v^2 + 2\rho(1-\rho^2)^{1/2} r_{uv} S_u S_v\right]^2}\right)\right]. \tag{3.4}$$

This can be done uniquely since $g(\rho; \mathbf{U}, \mathbf{V})$ is increasing in ρ, which is stated, but not proved, in

(Lillegard and Engen, 1999). The solution is

$$\tilde{\rho} = \frac{k}{\sqrt{1+k^2}},$$

where

$$k = \frac{S_{v^*}}{S_{u^*}} \left[\hat{\rho} - r_{u^* v^*} \left(\frac{1-\hat{\rho}^2}{1-r_{u^* v^*}^2} \right)^{1/2} \right] \left(\frac{1-\hat{\rho}^2}{1-r_{u^* v^*}^2} \right)^{-1/2}.$$

This way of generating values for $\tilde{\rho}$ is what they denote as conditional parametric bootstrapping in (Lillegard and Engen, 1999).

Now, what is the relationship between the true value ρ and a value $\tilde{\rho}$ generated from this procedure? In the paper they state that the rank of $g(\rho; \mathbf{U}, \mathbf{V})$ among the m generations of $g(\rho; \mathbf{U}^*, \mathbf{V}^*)$ is uniform on the integers $\{1, 2, \ldots, m+1\}$, because $(\mathbf{U}^*, \mathbf{V}^*)$ and $(\mathbf{U}, \mathbf{V})$ are from the same distribution. Further, if $g(t; \mathbf{U}, \mathbf{V})$ is an increasing function in t for any $\mathbf{U}, \mathbf{V}$, which the paper states that is often the case, then we have the following relationship

$$\{\tilde{\rho} < \rho\} = \{g(\rho; \mathbf{U}^*, \mathbf{V}^*) > g(\rho; \mathbf{U}, \mathbf{V})\}. \tag{3.5}$$

The reason for this is illustrated in Figure 3.1, which is a reconstructing of Fig. 1 in (Lillegard and Engen, 1999). First, assume that $g(\rho; \mathbf{U}, \mathbf{V}) > g(\rho; \mathbf{U}^*, \mathbf{V}^*)$, which is the case in the illustrating plot, where the blue line is above the orange line at $t = \rho$. We know that the maximum likelihood estimator is obtained by computing $\hat{\rho} = g(\rho; \mathbf{U}, \mathbf{V})$. The value of $\tilde{\rho}$ is found by solving $\hat{\rho} = g(\tilde{\rho}; \mathbf{U}^*, \mathbf{V}^*)$ for $\tilde{\rho}$. Now the question is whether or not $\tilde{\rho}$ is to the left of ρ. But because the blue line for $g(t; \mathbf{U}^*, \mathbf{V}^*)$ is above $g(t; \mathbf{U}, \mathbf{V})$ at $t = \rho$, and that this function is non-decreasing, we have that it cannot cross the horizontal line at $\hat{\rho}$ on the y-axis to the right of $t = \rho$. The only possible solution is the result in Equation (3.5), except for the possibility that $g(\rho; \mathbf{U}^*, \mathbf{V}^*) > \hat{\rho}$ for all values of t, but this is not investigated any further.

Looking pairwise at the true value of ρ and a simulated value $\tilde{\rho}$, we have from Equation (3.5) that the events $\{\tilde{\rho} < \rho\}$ and $\{\tilde{\rho} > \rho\}$ both have the same probability of 0.5 of happening when values for $(\mathbf{u}^*, \mathbf{v}^*)$ are generated. Hence, when m values for $\tilde{\rho}$ are generated we have that the number

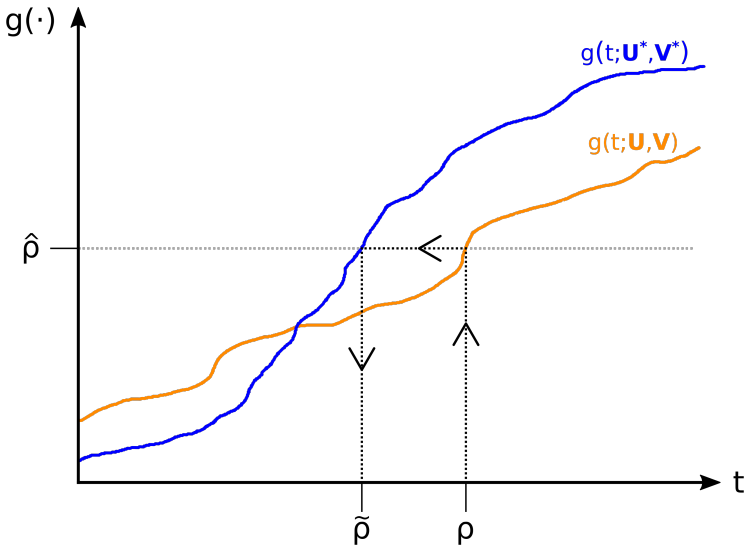

Figure 3.1: Illustration used to explain Equation (3.5).

of points that are below (or above) the true value of ρ is uniform on the integers $\{0,1,2,\ldots,m\}$, and hence the rank of ρ among the m values generated for $\tilde{\rho}$ is uniform on $\{1,2,\ldots,m+1\}$.

Now, to get an exact confidence interval we use that ρ has an equal probability to be inside any of the $m+1$ intervals between two successive numbers in the sorted list $(\tilde{\rho}_{(1)},\ldots,\tilde{\rho}_{(m)})$, where we also count the intervals $\left(-\infty,\tilde{\rho}_{(1)}\right)$ and $\left(\tilde{\rho}_{(m)},\infty\right)$. To construct a $(1-\alpha)100\%$ confidence interval, we can discard a proportion of $\frac{\alpha}{2}$ of the first intervals and $\frac{\alpha}{2}$ of the last intervals in the sorted list. The number of intervals k to remove on each side is then $k=\frac{\alpha}{2}(m+1)$, which means that the end points of the confidence interval is $\tilde{\rho}_{(k)}$ and $\tilde{\rho}_{(m-k+1)}$. For a given value of α, one must choose m so that k is an integer. For instance, if $\alpha=0.05$, then the possible values are $m=40l-1$ for $l\in\{1,2,\ldots\}$.

In summary, the algorithm is as follows:

Algorithm 1 - Conditional parametric bootstrapping on the bivariate normal distribution

Input: A random sample $(\mathbf{x}, \mathbf{y})$, confidence level $(1-\gamma)$ and number of simulations m.

1: Calculate $\hat{\rho}$ from $(\mathbf{x}, \mathbf{y})$ using Equations (3.1), (3.2) and (3.3).

Iterate m **times**:

2: Generate $\mathbf{U}^*$ and $\mathbf{V}^*$ independently from the univariate standard normal distribution.

3: Solve $\hat{\rho} = g(\tilde{\rho}; \mathbf{u}^*, \mathbf{v}^*)$ for $\tilde{\rho}$, where $g(\cdot)$ is defined in Equation (3.4).

End iteration

4: Sort the generated values: $(\tilde{\rho}_{(1)}, \ldots, \tilde{\rho}_{(m)})$.

5: Calculate $k = \frac{\gamma}{2}(m+1)$ and return the confidence interval $\left[\tilde{\rho}_{(k)}, \tilde{\rho}_{(m-k+1)}\right]$.

Figure 3.2 shows that the mean coverage of the generated confidence intervals converges to the desired value of 0.95. The left plot shows the coverage proportion as a function the logarithm (with base 2) of the number of iterations, the middle plots shows how the corresponding error decreases, while plot to the right shows the mean length of the generated confidence intervals. Here the values $n = 5$, $\rho = 0.5$, $\mu = [5, 10]^T$, $\sigma_x = 2$ and $\sigma_y = 3$ were used. The number of simulations in the paper was 500, but this was almost 20 years ago, so here we can run the simulations for $2^{20} \approx 1\,000\,000$ iterations.

In the plot the the left in Figure 3.2 we have included an approximate 0.95 confidence interval for each number of iterations, shown as vertical grey lines. This is to better evaluate the correctness of the algorithm. The height is calculated by noting that, for a number of iterations m, the number of values x inside the confidence interval will be binomially distributed with parameters m and $p = 0.95$, if the confidence interval is indeed exact. Hence, we can use the normal approximation $\frac{x}{m} \sim \mathrm{N}\left(p, \frac{p(1-p)}{m}\right)$, which will more than good enough for such large values of m when $p = 0.95$.

Figure 3.3 shows the same plots in the case of $n = 10$, with the same parameter values as with $n = 5$. Table 3.1 lists the final values for the coverage proportion and the mean length of the confidence intervals, using the largest number of iterations. These values are similar to the ones

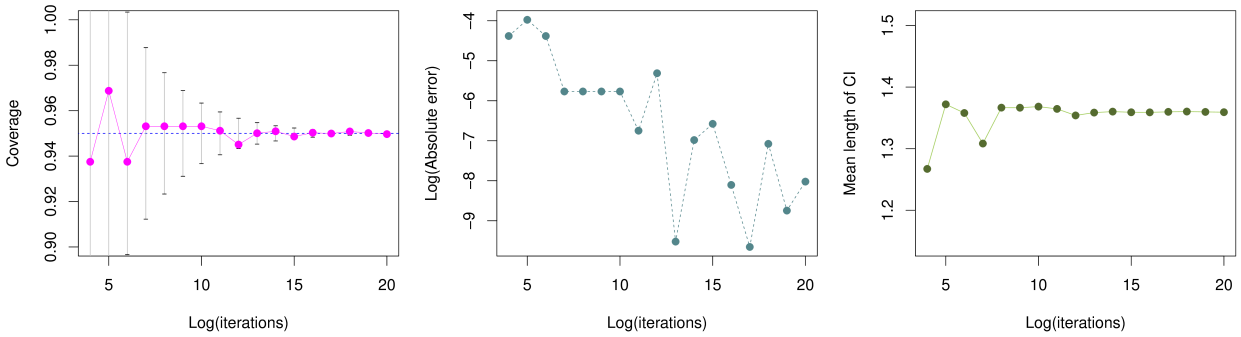

Figure 3.2: Calculated coverage proportion (left), corresponding error (middle) and mean length of confidence interval (right) of the correlation coefficient in the bivariate normal distribution, using conditional parametric boostrapping. Here the logarithm has base 2 and the sample size was $n = 5$ in each simulation.

listed in Table 1 in (Lillegard and Engen, 1999), but are of course closer to the desired coverage of 0.95, since the number of iterations is much larger. Looking at the two coverage proportion plots to the left in Figure 3.2 and 3.3, it looks like the coverage proportions are converging to 0.95 for both $n = 5$ and $n = 10$. Using the normal approximation described above, we calculate p-values of 0.240 and 0.372 for $n = 5$ and $n = 10$, respectively, for the coverage values at 2^{20} number of iterations, using the null hypothesis that the true coverage probability is 95%. We conclude that the method appears to be working fine in this example.

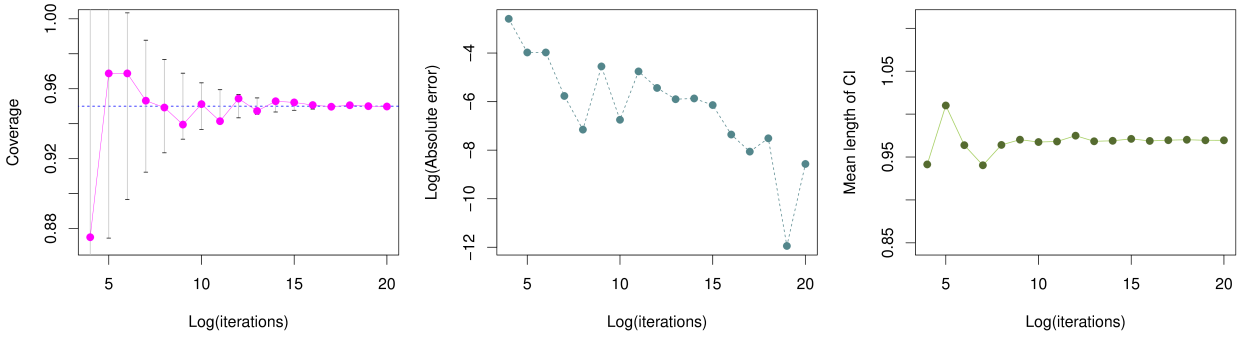

Figure 3.3: Calculated coverage proportion (left), corresponding error (middle) and mean length of confidence interval (right) of the correlation coefficient in the bivariate normal distribution, using conditional parametric boostrapping. Here the logarithm has base 2 and the sample size was $n = 10$ in each simulation.

n	Coverage proportion	Mean length
5	0.95025	1.3598
10	0.94981	0.9695

Table 3.1: Simulated coverage proportions and mean length of confidence intervals, using conditional parametric bootstrap on the bivariate normal distribution. Here the number of iterations used is $2^{20} \approx 1\,000\,000$.

3.2 Gamma Distribution

Now it's time to apply the method above to the gamma distribution. This is also done in (Lillegard and Engen, 1999), although the details are not given. Here we implement a method that is strongly motivated by the ideas and results in (Lindqvist and Taraldsen). Assume a random sample $\mathbf{x} = (x_1, \ldots, x_n)$ from the gamma distribution with parameters α and β, as given by Equation (2.1). Define two statistics as

$$T_1(\mathbf{X}) = \frac{1}{n}\sum_{i=1}^{n} X_i, \tag{3.6}$$

$$T_2(\mathbf{X}) = \frac{\left(\prod_{i=1}^{n} X_i\right)^{1/n}}{\frac{1}{n}\sum_{i=1}^{n} X_i}. \tag{3.7}$$

By looking at Equation (2.2) for the likelihood function of the gamma distribution, which is the same expression as for the joint density of the random sample, it is clear from the Factorization Theorem (see Section 2.4) that $T_1(\mathbf{X})$ and $T_2(\mathbf{X})$ are sufficient statistics. This can be seen by writing the functions in the Factorization Theorem as

$$\begin{aligned} h(\mathbf{x}) &= \left(\frac{1}{\Gamma(\alpha)\beta^{\alpha}}\right)^{n}, \\ g(T_1(\mathbf{x}), T_2(\mathbf{x}); \alpha, \beta) &= \left(\frac{\left(\prod_{i=1}^{n} X_i\right)^{1/n}}{\frac{1}{n}\sum_{i=1}^{n} X_i}\right)^{n(\alpha-1)} \left(\frac{1}{n}\sum_{i=1}^{n} X_i\right)^{n(\alpha-1)} \mathrm{e}^{-\frac{n}{\beta}\frac{1}{n}\sum_{i=1}^{n} x_i} \\ &= T_1(\mathbf{x})^{n(\alpha-1)} T_2(\mathbf{x})^{n(\alpha-1)} \mathrm{e}^{-\frac{n}{\beta}T_2(\mathbf{X})}. \end{aligned}$$

The conditional parametric bootstrap is here done by keeping the value of both T_1 and T_2 fixed, in the same way as correlation coefficient $\hat{\rho}$ was kept constant in the previous section.

Assume a random variable u is drawn from the uniform distribution from 0 to 1. Calculat-

ing

$$\chi(u;\alpha,\beta) = \beta \mathrm{F}^{-1}(u;\alpha,1), \tag{3.8}$$

where $\mathrm{F}^{-1}(u;\alpha,\beta)$ is the inverse of the cumulative distribution function $F(x;\alpha,\beta)$ of the gamma distribution with parameters α and β, we have by inversion that $\chi(u;\alpha,1) \sim \text{Gamma}(\alpha,1)$, and because β is a scale parameter, $\chi(u;\alpha,\beta) \sim \text{Gamma}(\alpha,\beta)$. So a random sample $\mathbf{x}$ can be thought of as having been generated in this way, with unknown values for the parameters α and β, and the random vector $\mathbf{u}$.

Now, Equations (3.6) and (3.7), which are functions of the random sample $\mathbf{x}$, can also be thought of as functions of the parameters α and β, and the random vector $\mathbf{u}$, using Equation (3.8). This gives the following equations:

$$g_1(\mathbf{u},\alpha,\beta) = \frac{\beta}{n}\sum_{i=1}^{n} \mathrm{F}^{-1}(u_i;\alpha,1), \tag{3.9}$$

$$g_2(\mathbf{u},\alpha) = \frac{\left(\prod_{i=1}^{n} \mathrm{F}^{-1}(u_i;\alpha,1)\right)^{1/n}}{\frac{1}{n}\sum_{i=1}^{n} \mathrm{F}^{-1}(u_i;\alpha,1)}. \tag{3.10}$$

In the conditional parametric bootstrap algorithm we need to solve the equations $T_1(\mathbf{x}) = g_1\left(\mathbf{u}^*,\tilde{\alpha},\tilde{\beta}\right)$ and $T_2(\mathbf{x}) = g_2(\mathbf{u}^*,\tilde{\alpha})$ for $\tilde{\alpha}$ and $\tilde{\beta}$, given values for $\mathbf{x}$ and a simulated random vector $\mathbf{u}^*$. Here $g_2(\mathbf{u}^*,\tilde{\alpha})$ is only a function of the first parameter $\tilde{\alpha}$, and hence we can solve this equation for $\tilde{\alpha}$ first, and use this solution to solve the second equation $T_1(\mathbf{x}) = g_1\left(\mathbf{u}^*,\tilde{\alpha},\tilde{\beta}\right)$ for $\tilde{\beta}$. This solution can easily be found explicitly for $\tilde{\beta}$.

In (Iliopoulos, 2016) it is shown that the function $g_2(\mathbf{u}^*,\tilde{\alpha})$ is increasing in $\tilde{\alpha}$ for all possible values of the simulated random vector $\mathbf{u}^*$. Hence, there is a maximum of one solution to the equation $T_2(\mathbf{x}) = g_2(\mathbf{u}^*,\tilde{\alpha})$. This solution is found in R by using the bisection method, which is a numerical root-finding method that is relatively easy to implement. The algorithm starts by specifying some initial values for two end points a and b and checks that $T_2(\mathbf{x}) - g_2(\mathbf{u}^*,a)$ and $T_2(\mathbf{x}) - g_2(\mathbf{u}^*,b)$ are of opposite signs. It not, either a is divided by two or b is multiplied by two until the signs are opposite. Then the bisection method works by calculating the function value at the mid-point $(a+b)/2$. If this value is less than the tolerance level specified, which in our

case was 10^{-8}, then the algorithm is done, if not, the algorithm continues with $(a+b)/2$ as one of the end-points, together with either a or b, depending on the signs of the calculated function values.

The conditional parametric bootstrap algorithm for the gamma distribution can be summarized as the following:

Algorithm 1 - Conditional parametric bootstrapping on the gamma distribution

Input: A random sample $\mathbf{x}$, confidence level $(1-\gamma)$ and number of simulations m.

1: Calculate $T_1(\mathbf{x})$ and $T_2(\mathbf{x})$ using Equations (3.6) and (3.7).

Iterate m **times**:

2: Generate $\mathbf{U}^* = (U_1^*, \ldots, U_n^*)$, where each U_i^* is drawn independently from the uniform distribution between 0 and 1.

3: Solve $T_2(\mathbf{x}) = g_2(\mathbf{u}^*, \tilde{\alpha})$ numerically for $\tilde{\alpha}$ using Equations (3.7) and (3.10). This is done using an implementation of the bisection method in R.

4: Solve $T_1(\mathbf{x}) = g_1(\mathbf{u}^*, \tilde{\alpha}, \tilde{\beta})$ for $\tilde{\beta}$ using Equations (3.6) and (3.9), after inserting the solution for $\tilde{\alpha}$ in the previous step.

End iteration

4: Sort the generated values: $\left(\tilde{\alpha}_{(1)}, \ldots, \tilde{\alpha}_{(m)}\right)$ and $\left(\tilde{\beta}_{(1)}, \ldots, \tilde{\beta}_{(m)}\right)$.

5: Calculate $k = \frac{\gamma}{2}(m+1)$ and return the confidence intervals $\left[\tilde{\alpha}_{(k)}, \tilde{\alpha}_{(m-k+1)}\right]$ and $\left[\tilde{\beta}_{(k)}, \tilde{\beta}_{(m-k+1)}\right]$.

3.3 Results and Discussion

The algorithm described in the previous section was tested out on the same two examples as in (Lillegard and Engen, 1999). The true value of the parameters was $(\alpha, \beta) = (0.5, 1)$ in both cases, while the number of data points in the random sample $\mathbf{x}$ was chosen to be $n = 5$ and $n = 10$. The resulting convergence proportions after running the simulations 2^{21} times for $n = 5$ and 2^{20} times for $n = 10$, are shown in the last column of Table 3.2. The number of simulations m at each iteration in the algorithm above, was chosen to be $m = 199$, corresponding to a cut-off value of

$k = 5$. The confidence level used was $(1-\gamma) = 0.95$.

Table 3.2 also shows the results when using the parametric and non-parametric bootstrap methods, which are described in Section 2.3. Here the estimators used in both cases are the maximum likelihood estimators, obtained using the fitdistr-function from the MASS-library in R. Repeated calculations for the value of these estimators are done for 1 000 bootstrap samples, and used to calculate approximate 95%-confidence intervals. Clearly, these types of bootstrap methods provide confidence intervals with a coverage proportion that differs quite much from the nominal coverage of 0.95. The main reason for this is the fact that the distribution F^* is a poor approximation of the true distribution F for such small values of n. The conditional parametric bootstrap works better, and the coverage proportions for both α and β are quite close to 95%. The convergence of the coverage proportions are investigated below.

Table 3.2: Simulated coverage proportions of confidence intervals for the parameters in the gamma distribution.

(n, α, β)	Parameter	Par. Boostrap	Non-par. Bootstrap	Conditional Par. Bootsrap
(5,0.5,1)	α	0.739	0.616	0.95006
	β	0.694	0.523	0.95111
(10,0.5,1)	α	0.825	0.769	0.94965
	β	0.804	0.718	0.95015

Table 3.3 shows the mean length of the confidence intervals. Clearly, the conditional parametric bootstrap algorithm generates confidence intervals that are much narrower than those produced by regular bootstrapping, especially for the smallest value $n = 5$. The results in Tables 3.2 and 3.3 can be compared with the results in Table 2.1 in (Lillegard and Engen, 1999).

The intervals from the non-parametric bootstrap are very wide, and this is because when n is small there is a small probability of obtaining a sample x^* of very similar values when doing the bootstrap resampling from the empirical distribution. For instance, generating a random sample of size $n = 5$ from the gamma distribution with parameters $(\alpha, \beta) = (0.5, 1)$ gave in one

case the following result:

$$x = [5.0468, 0.2037, 0.2016, 0.0013, 1.0310].$$

One possible resample of this could be

$$x^* = [0.2037, 0.2016, 0.2037, 0.2037, 0.2037].$$

Calculating the maximum likelihood estimators for the parameters using this resample x^* and the fitdistr-function from the MASS-library in R, we get

$$\hat{\alpha} = 44\,711.05$$
$$\hat{\beta} = 219\,922.98,$$

which is extremely far from the true values of $\alpha = 0.5$ and $\beta = 1$. Although such resamples will be quite rare, it is probably the effect of these that makes the confidence intervals so wide when using non-parametric bootstrap on samples of size $n = 5$.

Table 3.3: Simulated mean lengths of confidence intervals for the parameters in the gamma distribution.

(n, α, β)	Parameter	Par. Boostrap	Non-par. Bootstrap	Conditional Par. Bootsrap
(5,0.5,1)	α	9.30	282.8	2.03
	β	42.54	3555.2	8.30
(10,0.5,1)	α	1.77	1.94	0.94
	β	7.35	9.14	3.28

Figure 3.4 and 3.5 shows the coverage proportions for the confidence intervals of α and β, respectively, in the case when $n = 5$. The vertical lines are here the same approximated 95% confidence intervals for the coverage proportions used earlier in the case of the bivariate normal distribution, under the assumption that the true coverage probability for this method is indeed 95%. Figures 3.6 and 3.7 shows similar plots, only now for the case of $n = 10$. Here the logarithm used is in base 2. From these plots we can note the following:

- The coverage proportion of the parameter α seems to be converging towards 95% for both $n = 5$ and $n = 10$.
- The coverage proportion of the parameter β, however, is not converging to the nominal coverage probability of 95%. This is most clear from the plot to the right in Figure 3.5, where the last three points are far outside the vertical bars indicating the approximate 95% confidence intervals at each number of iterations. There are two plausible reasons for this:
    1. The numerical calculation of $\tilde{\alpha}$ using the bisection method introduces some error that effects the results.
    2. The conditional parametric bootstrap is in fact not an exact method for this application.

Because the values for $\tilde{\alpha}$ are found numerically by itself, and then used in the equation that computes $\tilde{\beta}$, it seems unlikely that a numerical inaccuracy would effect the coverage proportion for β much more significantly than the coverage proportion for α. It can, however, not be ruled out totally.

In (Lillegard and Engen, 1999) they state, when talking about the gamma distribution, that "... the ordered bootstrap replicates produce intervals with the exact cover probability for each parameter considered seperately.". From the simulation studies done in this thesis is seems that this statement might be wrong, and this is also suspected in (Lindqvist and Taraldsen).

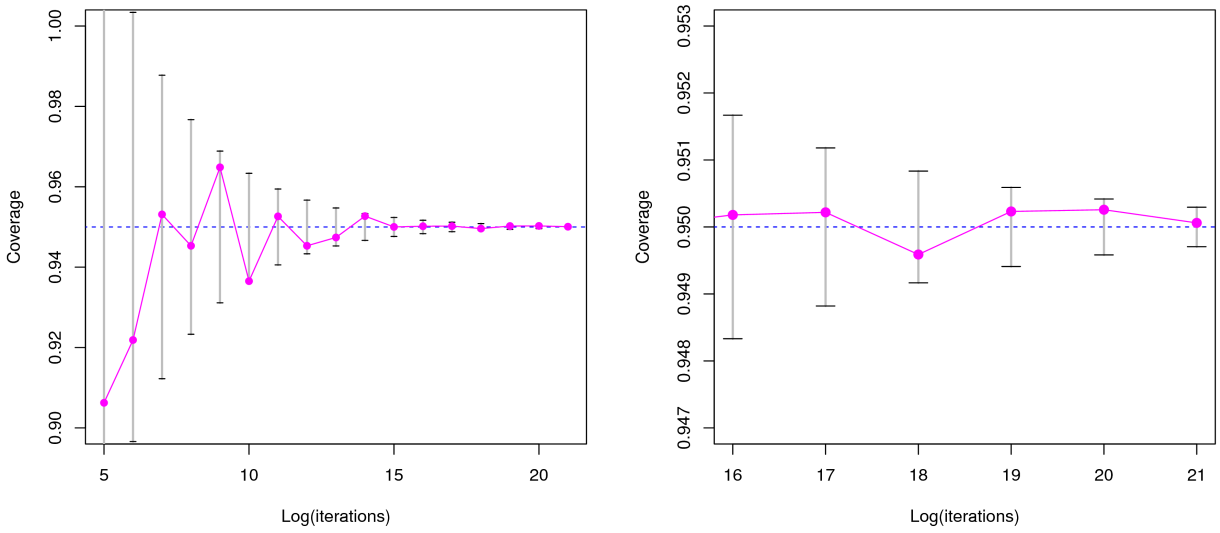

Figure 3.4: Convergence proportion for the confidence interval of alpha, as a function of the logarithm (with base 2) of the number of iterations. Here the sample size is $n = 5$ and the true values of the parameters are $\alpha = 0.5$ and $\beta = 1$. The right plot zoomes in on the last 5 values.

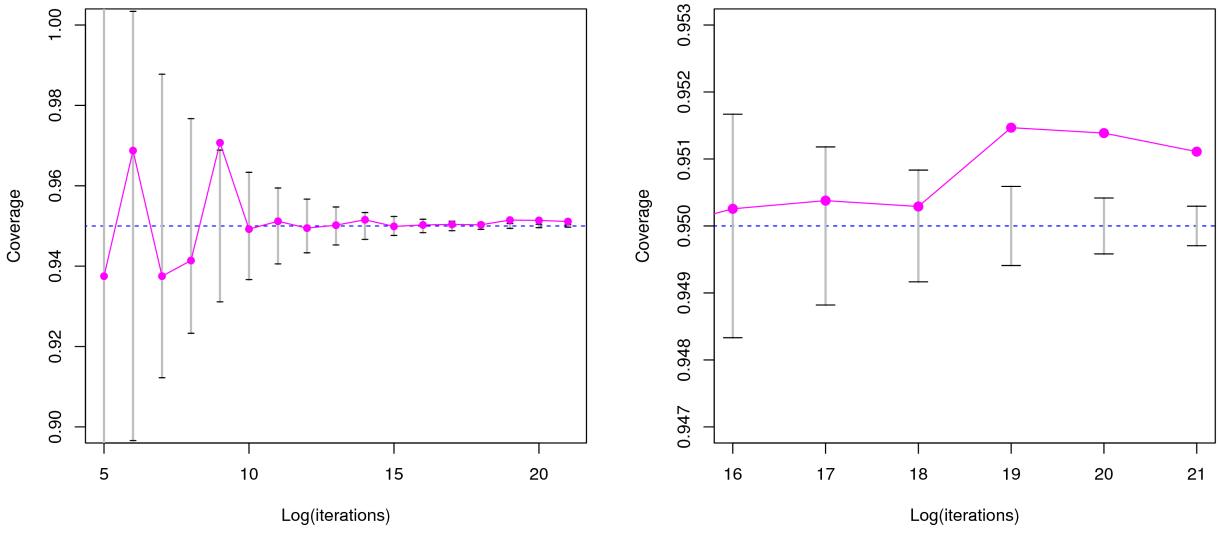

Figure 3.5: Convergence proportion for the confidence interval of beta, as a function of the logarithm (with base 2) of the number of iterations. Here the sample size is $n = 5$ and the true values of the parameters are $\alpha = 0.5$ and $\beta = 1$. The right plot zoomes in on the last 5 values.

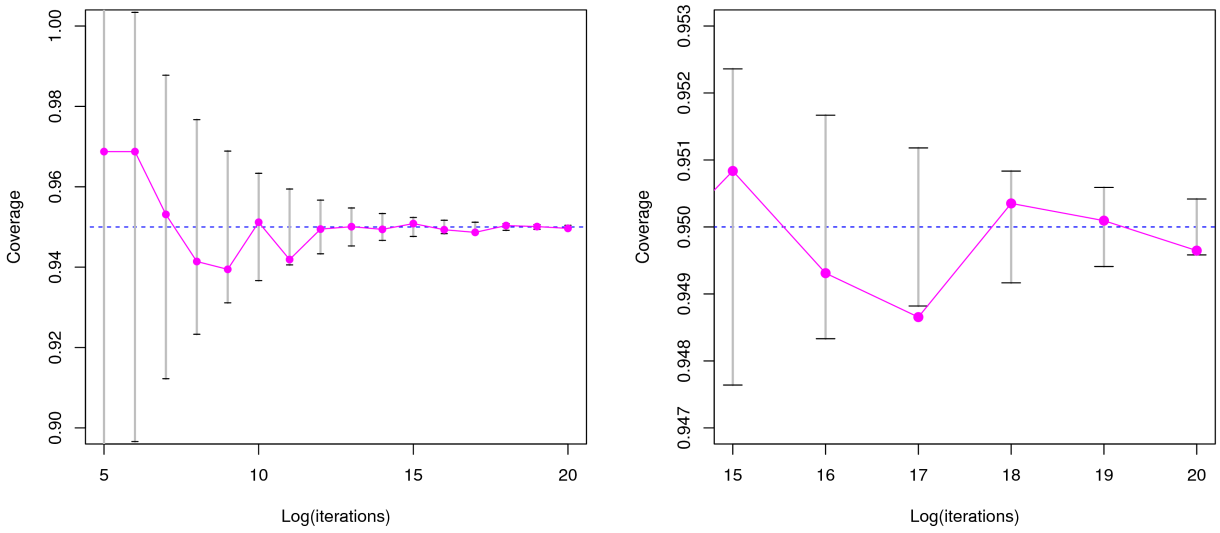

Figure 3.6: Convergence proportion for the confidence interval of alpha, as a function of the logarithm (with base 2) of the number of iterations. Here the sample size is $n = 10$ and the true values of the parameters are $\alpha = 0.5$ and $\beta = 1$. The right plot zoomes in on the last 5 values.

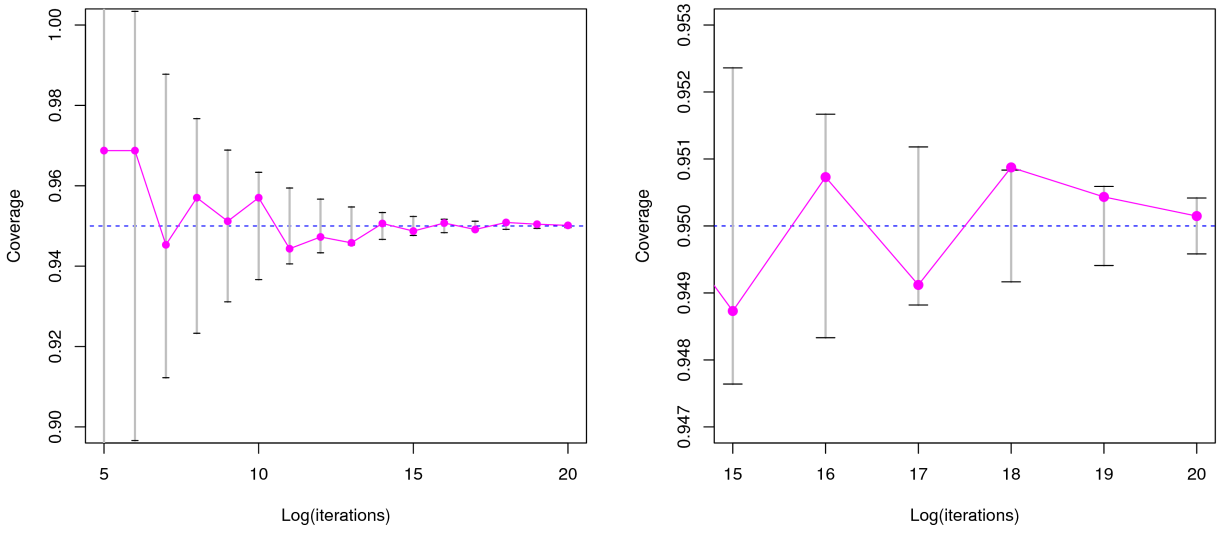

Figure 3.7: Convergence proportion for the confidence interval of beta, as a function of the logarithm (with base 2) of the number of iterations. Here the sample size is $n = 10$ and the true values of the parameters are $\alpha = 0.5$ and $\beta = 1$. The right plot zoomes in on the last 5 values.

Chapter 4

Probability Regions

Now we take a different approach to the inference of the parameters in the gamma distribution, namely to look at one confidence region in $\mathbb{R}^2$ for both α and β simultaneously, instead of looking at two confidence intervals in one dimension. To do this, it helps to have some notion of how to order data in two dimensions. We start by looking briefly at the one-dimensional case, with the usual ordering of values from smallest to largest.

In this chapter we will denote the regions and intervals constructed as "probability regions" or "probability intervals". They are in a sense prediction regions (or intervals), since they give a likely region for the location of a new generated point, but the the probability for this point to be inside the region is set before a random sample is drawn. This means that after such a random sample is generated, the probability for the new point to be included in the region will depend on the values of the actual data sample and not have the desired probability of covering this new point. Prediction regions (and intervals) are more difficult to construct, because in that case you want to have a certain probability for the new generated value to be included in the region, given the actual values of the data sample. For the probability intervals and regions discussed here, all we know is that there is a certain probability (or an approximate probability) for this generated point to be inside the sample, before the random sample itself has been generated.

This seems to be a strange way of generating intervals and regions, but we will see in Chapter 5 that we can combine methods from Chapter 3 with such types of probability regions to construct

confidence regions for the parameters in the gamma distribution.

We will denote the desired probability for a future sample point to be inside the probability region or interval, as the coverage probability level.

4.1 Order Statistics - One Dimension

Assume n random variables $X_1, \ldots, X_n$ are independent and identically distributed from a continuous univariate probability distribution $f(x)$, and that the ordered sample has the usual notation

$$X_{(1)}, \ldots, X_{(n)}.$$

If we generate one more random variable from the same distribution, $X_{n+1} \sim f(x)$, independent of the original values $X_1, \ldots, X_n$, then the placement of this value in the ordered sample of size $n+1$ will have probability $\frac{1}{n+1}$ for all possible placements $X_{(i)}, i = 1, \ldots, n+1$. This is the same as saying that X_{n+1} is distributed uniformly among the $n+1$ intervals between the values of $X_{(1)}, \ldots, X_{(n)}$, where also the end intervals $(-\infty, X_{(1)})$ and $(X_{(n)}, \infty)$ are included. This is illustrated in Figure 4.1, showing the location of five samples points and the possible intervals where a new random variable might lie. Of course, the probability distribution for X_{n+1} over these $n+1$ possible intervals will change if conditioned on the actual values $x_{(1)}, \ldots, x_{(n)}$ generated.

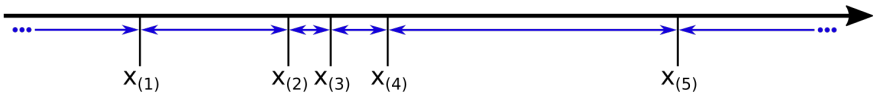

Figure 4.1: An ordered sample of size 5 from a continuous probability distribution. The blue arrows indicates the 6 intervals where a future sample point may lie.

This is tested for the following five different continuous univariate distributions:

- $X_i \sim N(\mu, \sigma^2)$ with $\mu = 2$ and $\sigma = 0.3$.
- $X_i \sim \text{Exp}(\beta)$ with rate parameter $\beta = 2$.

- $X_i \sim \text{Unif}[a,b]$ with $a = 0$ and $b = 1$.
- $X_i \sim \text{Gamma}(\alpha,\beta)$ with shape parameter $\alpha = 3$ and rate parameter $\beta = 0.5$.
- $X_i = U_i \cos(\Theta_i)$, where $U_i \sim \text{Unif}[0,1]$ and $\Theta_i \sim \text{Exp}(\beta)$, with rate parameter $\beta = 3$, denoted "Test5".

In each case the sample size is $n = 9$, so that X_{10} should be uniformly distributed over the ten possible intervals. Figure 4.2 shows the results after doing this for three different number of simulations; 100 (left), 10 000 (middle) and 1 000 000 (right). Each vertical bar shows the proportion of times that the generated value x_{10} lies inside each of the ten possible interval from a sorted random sample $\left(x_{(1)},\ldots,x_{(9)}\right)$. It seems clear that in all five cases the proportion of times X_{n+1} falls into each one of the 10 different intervals converges towards $\frac{1}{10}$, as it should.

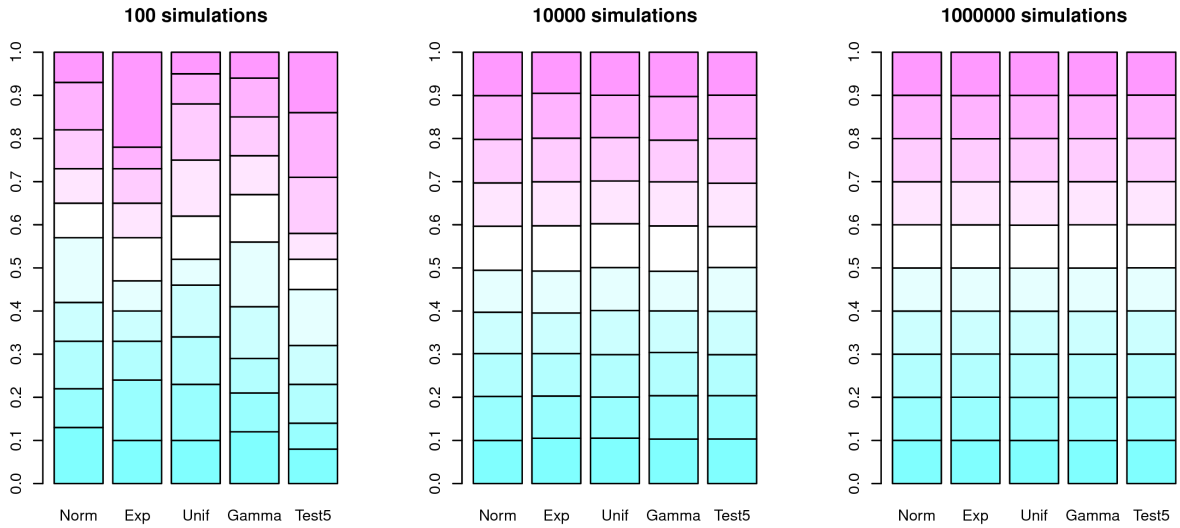

Figure 4.2: Proportion of times x_{10} falls into each of the 10 different intervals from a sorted random sample $\left(x_{(1)},\ldots,x_{(9)}\right)$, for three different number of simulations. Here each vertical bar shows the proportion of times that x_{10} falls in each of the 10 possible intervals, starting with $\left(-\infty,x_{(1)}\right)$ at the bottom and ending with $\left(x_{(9)},\infty\right)$ on the top. The sum of all proportions is of course equal to 1 in all cases. The text under each bar shows what the underlying distribution is.

4.2 Order Statistics - Two Dimensions

Now one might ask, is it possible to do something similar in two dimensions? More precisely, if $(X_1, Y_1), \ldots, (X_n, Y_n)$ are independent and identically distributed from a continuous bivariate probability distribution $f(x, y)$, is it possible to partition the two-dimensional domain of the random variables such that a new generation (X_{n+1}, Y_{n+1}) have an equal probability to be inside each of the areas from the partition? In the investigation of this we use two different bivariate probability distributions:

- The bivariate normal distribution $(\mathbf{X}, \mathbf{Y}) \sim N_2(\boldsymbol{\mu}, \Sigma)$, with parameters

$$\boldsymbol{\mu} = \begin{bmatrix} 5 \\ 3 \end{bmatrix} \quad \text{and} \quad \Sigma = \begin{bmatrix} \sigma_x^2 & \rho\sigma_x\sigma_y \\ \rho\sigma_x\sigma_y & \sigma_y^2 \end{bmatrix} = \begin{bmatrix} 3 & 0.7 \\ 0.7 & 0.8 \end{bmatrix}. \tag{4.1}$$

- The random vector $(\mathbf{X}, \mathbf{Y})$ with $(X_i, Y_i) = (U_i \cos(\Theta_i), U_i \sin(\Theta_i))$, $i = 1, \ldots, n$, where $U_i \sim \text{Unif}[0, 1]$ and $\Theta_i \sim \text{Exp}(\beta)$, with rate parameter $\beta = 1.6$. From here on denoted as the "UnitCircle-distribution".

A contour plot of the density of the bivariate normal distribution is shown to the left in Figure 4.3, together with a plot of 200 000 generated values from the UnitCircle-distribution shown to the right, to illustrate how that density looks. The UnitCircle-density is zero for all points outside the unit circle, and have a quite complicated shape compared to the bivariate normal density. For instance, the straight line from origo to $(x, y) = (1, 0)$ seperates a high-density region from a low-density region. One might except that it in this case will be harder to construct sensible probability regions with an approximately correct coverage probability, compared to when the data sample is from the bivariate normal distribution.

One possibility might be that if one partitions the two-dimensional domain by drawing vertical and horizontal lines through all the sample points, then all the regions generated will have an equal probability of $\frac{1}{(n+1)^2}$ of containing the next value X_{n+1}. It turns out that this is not the case. This was shown for the bivariate normal distribution and for the UnitCircle-distribution, but we omit the details here.

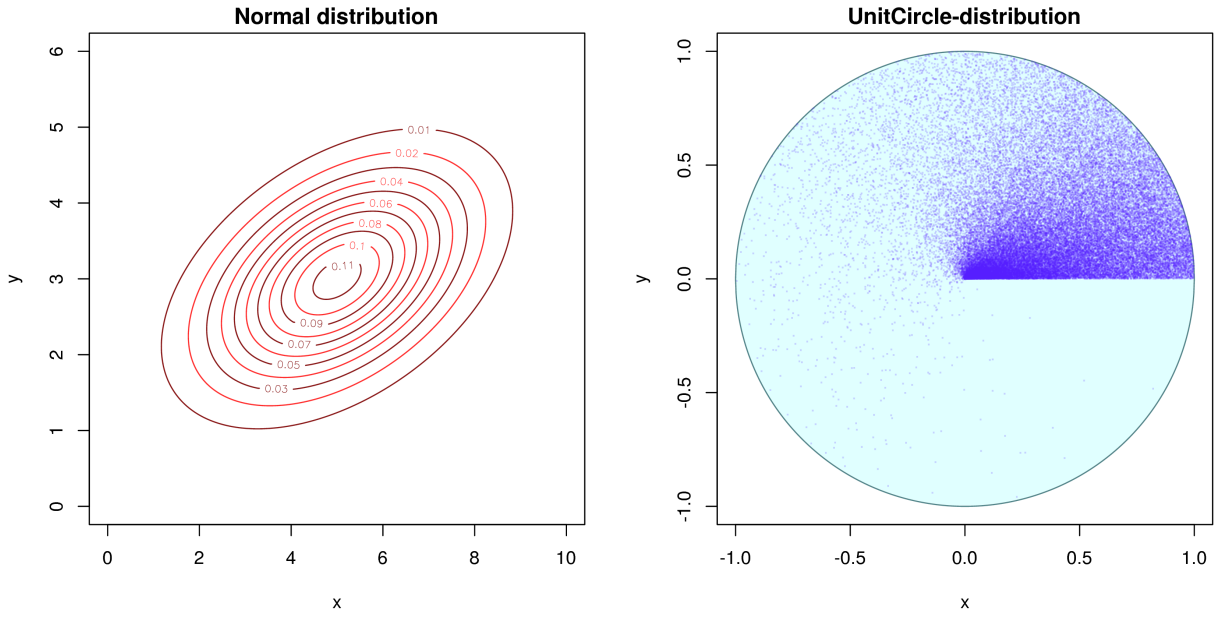

Figure 4.3: Left: A contour plot of the bivariate normal density. Right: A plot of 200 000 generated values from the UnitCircle-distribution. The region inside the unit circle is colored in light blue, and shows where the probability density function is non-zero.

A different approach is to find a way to sort the sample points, and construct a probability region using this sorted list. More precisely, define a function $g : \mathbb{R}^2 \to \mathbb{R}$ and sort the values of $g(X_1, Y_1), \ldots, g(X_n, Y_n)$. If the pairs (X_i, Y_i) are independent and indentically distributed for all $i = 1, \ldots, n$, then $g(X_1, Y_1), \ldots, g(X_n, Y_n)$ is just a sample of independent and identically distributed random variables, and hence the ordering of the sorted sample will be uniform, as was stated in the previous section. For instance, one could choose the function $g(x, y) = \sqrt{x^2 + y^2}$, which is just the distance from origo. But, in general we have no way of knowing if the sample points should lie close to origo or not. The probability region constructed using this approach will in general be an annulus, because small and big values for the distance will be disregarded (or just one of the two, if a one-sided probability interval of the sorted values is preferred). The area of such a probability region will probably be much larger than what is necessary, depending on the distribution of the sample points, because the region might cover large regions where the density of points is small. So this is not a good choice in general.

A sensible choice for the function $g(x, y)$ should order the sample in such a way that the "most extreme" sample points will be either in the beginning or at the end of the sorted list. To do

this, we must have some way of defining what "most extreme" means in $\mathbb{R}^2$. Look for instance at Figure 4.4, which shows an example of a random sample. Here it is clear that there is a dense cloud in the middle, where most of the points lies, and that some outliers are located quite far away from this "center", which is roughly what we mean by "most extreme". This is the notion of data depth, which is a function that takes a data cloud $(x_1, y_1), \ldots, (x_n, y_n)$, or alternatively the corresponding underlying distribution F in $\mathbb{R}^2$, and calculates a depth value that indicates how "deep" the sample point is inside the data cloud. There are many different types of data depth, and some of them will be discussed below.

In (Zuo and Serfling, 2000) they suggest what properties a depth function should have in general, and many of the well known depth functions are classified according to this. The four key properties proposed in the article are roughly the following:

① **Affine invariance**: The depth value at a particular point is independent on the underlying coordinate system.

② **Maximality av center**: The depth function obtains its maximum at the center of the underlying distribution F, if such a center exists.

③ **Monotonicity relative to deepest point**: The depth decreases monotonically as you move away from the deepest point (the point with the highest depth) along any straight line.

④ **Vanishing at infinity**: The depth function approaches zero as you move infinitely away from origo.

Note that these properties are for the depth function and doesn't necessarily hold for the sample depth function.

4.3 Depth Statistics

Here we introduce various depth functions. We will use $D_{\ldots}(x; F)$ to denote the depth at point x, where the data comes from some underlying distribution F, and $SD_{\ldots}(x; \boldsymbol{\omega})$ to denote the sample depth at point x with respect to the data cloud $\boldsymbol{\omega} = (\omega_1, \ldots, \omega_n)$.

Figure 4.4: An example of a random sample in $\mathbb{R}^2$.

4.3.1 Mahalanobis Depth

One popular choice for data depth is the Mahalanobis depth, introduced in (Mahalanobis, 1936). The Mahalanobis depth at point x that has an underlying distribution F, is defined as

$$D_{\text{mal}}(x;F) = \left[1 + (x-\mu_F)^T \Sigma_F^{-1} (x-\mu_F)\right]^{-1},$$

where μ_F and Σ_F is the mean vector and covariance matrix of F, respectively. See for instance (Liu and Singh, 1997). The sample version of the Mahalanobis depth at point x with respect to a set of data points $\boldsymbol{\omega} = (\omega_1, \ldots, \omega_n)$ is defined as

$$SD_{\text{mal}}(x;\boldsymbol{\omega}) = \left[1 + (x-\bar{\boldsymbol{\omega}})^T \hat{\Sigma}_{\boldsymbol{\omega}}^{-1} (x-\bar{\boldsymbol{\omega}})\right]^{-1}, \tag{4.2}$$

where $\bar{\boldsymbol{\omega}}$ is the mean of ω and $\hat{\Sigma}_{\boldsymbol{\omega}}$ is the empirical covariance matrix. In our case both x and the ω_i's will be in $\mathbb{R}^2$. The range of this function goes from 0 (for points infinitely away from the mean) and up to 1 (for a points exactly at the mean). Figure 4.5 illustrates what the sample Mahalanobis depth looks like for data generated from the bivariate normal distribution, with parameters given by Equation (4.1). The left plot shows a data set of size $n = 5$, while the right plot shows a data set of size $n = 15$. In both cases we observe that the depth obtains its highest

value around the center of the data cluster, and gradually decrease as we move away from this center. The contours of the Mahalanobis depth will always be ellipses (see details below).

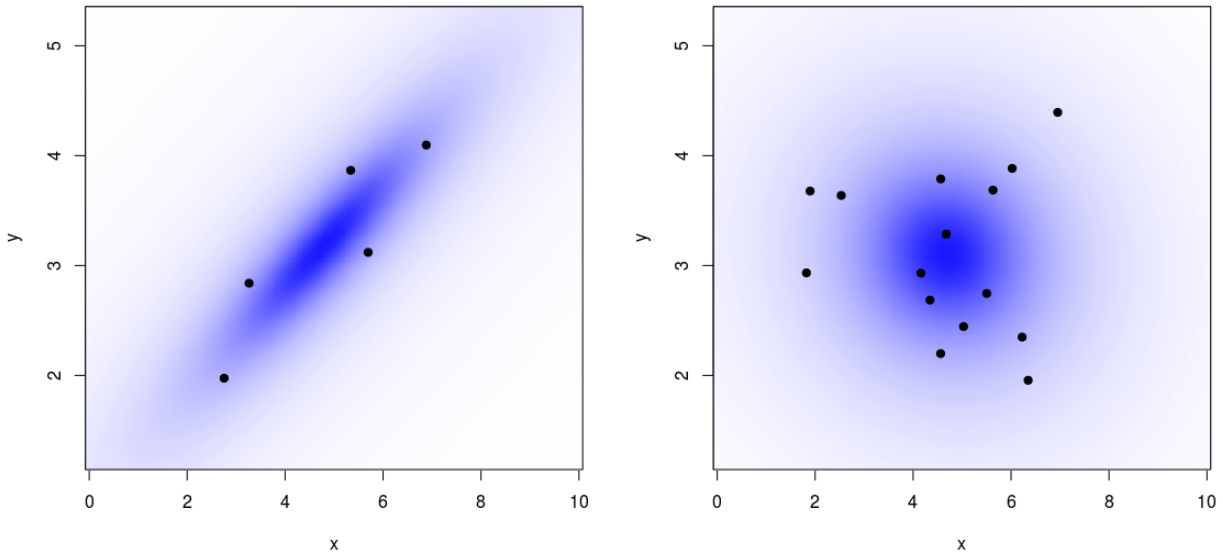

Figure 4.5: Illustration of the sample Mahalanobis depth, using data generated from the bivariate normal distribution, with sample sizes $n = 5$ (left) and $n = 15$ (right). A dark color corresponds to a larger value of the depth.

4.3.2 Simplicial Depth

The simplicial depth of a point x with respect to a distribution F was introduced in (Liu, 1990), which use the concept of a simplex. If we restrict ourselves to $\mathbb{R}^2$, then a simplex is simply a triangle. The simplicial depth $D_{\text{sim}}(x;F)$ of a point x is then the probability that x will fall inside the triangle made up by three independent sample points X_1, X_2, X_3 from a distribution F. This can be written as

$$D_{\text{sim}}(x;F) = \mathrm{P}_F(x \in \Delta(X_1, X_2, X_3)),$$

where $\Delta(\cdot)$ denotes the set of numbers inside the triangle (the closed set). The sample simplicial depth $SD_{\text{sim}}(x, \boldsymbol{\omega})$ of a point x with respect to a data cloud $\boldsymbol{\omega} = (\omega_1, ..., \omega_n)$, is defined as

$$SD_{\text{sim}}(x, \boldsymbol{\omega}) = \frac{1}{\binom{n}{3}} \sum_{1 \le i < j < k \le n} \mathrm{I}\left(x \in \Delta(\omega_i, \omega_j, \omega_k)\right),$$

which is just the proportion of all possible triangles, formed by the cloud of points, that contains x. Note that the edges, and therefore also the vertices, of the triangle are always included in the set $\Delta(\cdot)$, meaning that $\omega_1, \omega_2, \omega_3$ are all counted as being in the triangle $\Delta(\omega_1, \omega_2, \omega_3)$. Figure 4.6 illustrates what the sample simplicial depth looks like for data generated from the bivariate normal distribution, by drawing and coloring all the $\binom{n}{3}$ possible triangles for a sample of size $n = 5$ at the left and a sample of size $n = 15$ to the right. Note that these data samples are the same ones used in Figure 4.5 for the illustration of the sample Mahalanobis depth. Darker color means higher value for the depth. It is clear that the depth is highest around the "center" of the data cloud, and that the depth is zero for all points outside the convex hull of all sample points (the definition of a convex hull is presented later in Section 4.4.3). Also notice that the picture looks more complicated for higher values of n. This is simply because the number of triangles is much larger. This number, $\binom{n}{3}$, is approximately equal to $\frac{n^3}{6}$ when n is large, and so it grows quite rapidly.

4.3.3 Adjusting the Simplicial Depth

If F is an absolutely continuous probability distribution, then the simplicial depth $D_{\text{sim}}(x; F)$ will be a continuous function (see (Liu, 1990), Theorem 2). The sample version $SD_{\text{sim}}(x; \boldsymbol{\omega})$ will of course not be a continuous function. It turns out that the sample simplicial depth at ω_a, where ω_a is one of the sample points $\omega_1, \ldots, \omega_n$ used to calculate $SD_{\text{sim}}(x; \boldsymbol{\omega})$, has a value that differs some from the depth values of surrounding points. To see this, one can divide the calculation of the depth value at point ω_a into two parts: the first consisting of triangles that do not use ω_a as one of the three vertices of the triangle, and the second where ω_a is one the three vertices. We

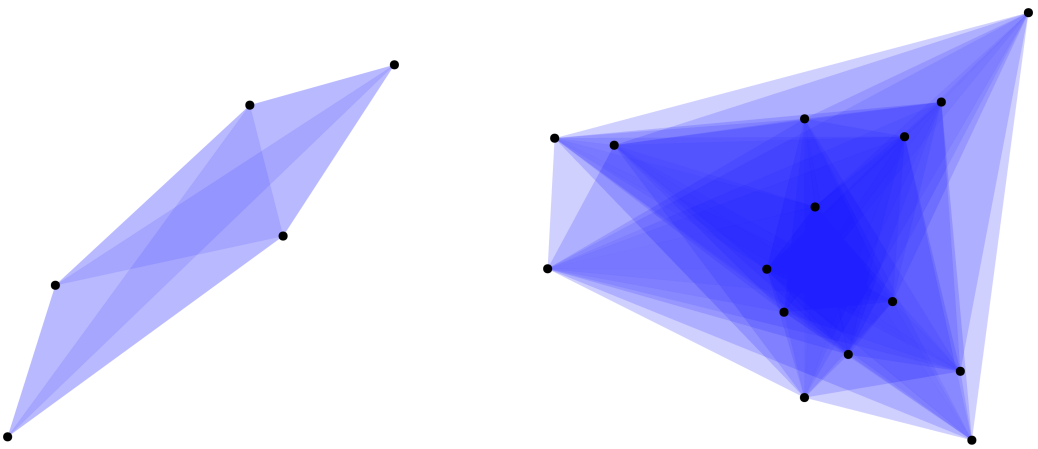

Figure 4.6: Illustration of the sample simplicial depth, using data generated from the bivariate normal distribution, with sample sizes $n = 5$ (left) and $n = 15$ (right). A dark color means that many triangles include that area, corresponding to a larger value of the depth. When $n = 5$ there are 10 unique triangles, while when $n = 15$ there are 455 unique triangles.

can write this as

$$SD_{\text{sim}}(\omega_a; \boldsymbol{\omega}) = \frac{1}{\binom{n}{3}} \left(\sum_{\substack{1 \le i < j < k \le n \\ i,j,k \neq a}} \mathrm{I}\left(\omega_a \in \Delta(\omega_i, \omega_j, \omega_k)\right) + \sum_{\substack{1 \le i < j < k \le n \\ i,j \text{ or } k = a}} \mathrm{I}\left(\omega_a \in \Delta(\omega_i, \omega_j, \omega_k)\right) \right). \quad (4.3)$$

The indicator function in the last term will be equal to one for all the terms in the sum, since $\Delta(\cdot)$ is a closed set. The number of triangles formed by ω_a as one of the vertices is $\binom{n-1}{2}$, and so we can, after some calculations, write

$$SD_{\text{sim}}(\omega_a; \boldsymbol{\omega}) = \frac{1}{\binom{n}{3}} \sum_{\substack{1 \le i < j < k \le n \\ i,j,k \neq a}} \mathrm{I}\left(\omega_a \in \Delta(\omega_i, \omega_j, \omega_k)\right) + \frac{3}{n}. \quad (4.4)$$

Here $\frac{3}{n}$ is the proportion of triangles that has ω_a as one of the vertices.

Now, if we look at a point that lies close to ω_a, how is the depth at this point calculated? Let's say this new point is $\omega_a + \varepsilon$, where ε is a two-dimensional vector, and $|\varepsilon|$ is small (since n is finite, the simplicial depth will be a constant value inside a finite number of areas, and so $|\varepsilon|$ can be chosen to be small enough so that the depth value is constant for all vectors $\omega_a + \eta$ when η has the same

direction as ε and $0 < |\eta| \leq |\varepsilon|$). The depth at $\omega_a + \varepsilon$ will, with this choice of ε, have the same value for the first term in Equation (4.4), where the sum is over all the triangles not composed of ω_a as one of the three vertices. The second term, however, will almost certainly be different. Among all the triangles that involves ω_a, only some of them will cover $\omega_a + \varepsilon$. The effect of this is illustrated in Figure 4.7, which shows the sample simplicial depth for two different samples of size 5. The depth at points outside the shaded areas are 0. The simplicial depth at the five sample points are printed in purple. These values are actually larger than the depth values at all other points in $\mathbb{R}^2$, and doesn't really represent well the depth at surrounding regions. The values printed in green will be explained below.

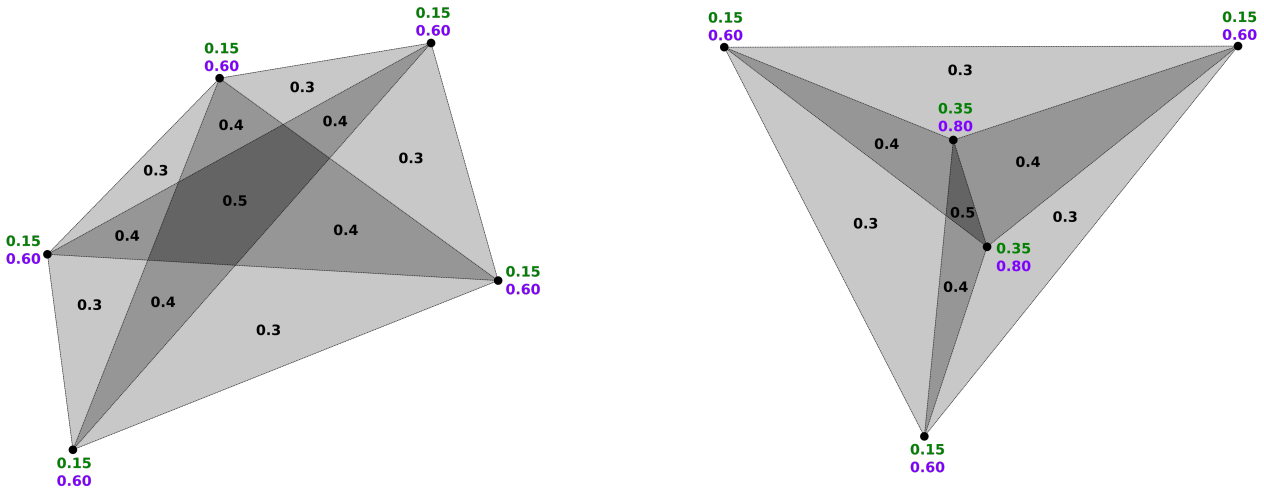

Figure 4.7: Sample simplicial depth for two data sets of size 5. More shading corresponds to a higher depth. The depth value at the sample points are printed in purple, while adjusted values are printed in green.

Is there any way to adjust the simplicial depth at sample points in such a way that it better represents the depth at points surrounding it? In Figure 4.8 we have zoomed in on a sample point ω_a, shown in blue. The degree of shading of the areas around indicates the values of the depth, just like in the previous figure. All the lines originating from ω_a is connected to some other sample points. It is clear that no matter what direction ε has, the new point $\omega_a + \varepsilon$ will be outside many of the triangles formed by ω_a as one of the vertices.

How many triangles can we expect to be covering the point $\omega_a + \varepsilon$? This will of course depend on the distribution of the sample points. Let's look at a simplified case where all the directions to the $n-1$ other sample points are uniformly distributed between 0 and 2π. For any two sample points

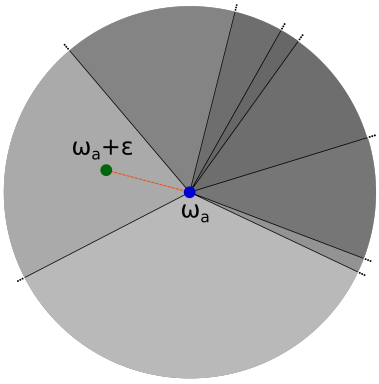

Figure 4.8: A sample point ω_a shown in blue, with shading around that corresponds to the value of the sample simplicial depth. The lines originating from ω_a are connected to other sample points. The green point shows a point $\omega_a + \varepsilon$, that lies a small distance $|\varepsilon|$ away from ω_a.

ω_i and ω_j, we then have that the angle between the line from ω_a to ω_i and the line from ω_a to ω_j, is uniformly distributed between 0 and π. Here we always choose the angle that is less than or equal to π, because this is the value of the corresponding angle in the triangle $\Delta(\omega_a, \omega_i, \omega_j)$ formed by the three points ω_a, ω_i and ω_j. If the direction of $\omega_a + \varepsilon$ is drawn from the uniform distribution between 0 and 2π, it follows that this point has a probability of $\frac{1}{4}$ of being between the two lines, or equivalently being inside the triangle $\Delta(\omega_a, \omega_i, \omega_j)$. Note that we here assume that $|\varepsilon|$ is so small that the first term in Equation (4.3) is constant for all possible directions of ε. Denote V_{ij} as the binary random variable that is equal to 1 if $\omega_a + \varepsilon$ is inside $\Delta(\omega_a, \omega_i, \omega_j)$, and 0 if it's not. Define V as the random variable that counts the total number of triangles, with ω_a as one of the vertices, containing the point $\omega_a + \varepsilon$. We can then compute the expected value as

$$\begin{aligned}
E[V] &= E\left[\sum_{i=1}^{n-1}\sum_{j=i+1}^{n-1} V_{ij}\right] \\
&= \sum_{i=1}^{n-1}\sum_{j=i+1}^{n-1} E\left[V_{ij}\right] \\
&= \sum_{i=1}^{n-1}\sum_{j=i+1}^{n-1} P\left(V_{ij}=1\right) \\
&= \sum_{i=1}^{n-1}\sum_{j=i+1}^{n-1} \frac{1}{4} \\
&= \frac{1}{4}\binom{2}{n-1}.
\end{aligned}$$

Hence, the expected number of triangles is $\frac{1}{4}$ times the total number of triangles. This expected value will of course be a bit different if the sample points around ω_a have a more complicated distribution.

Here we propose to adjust the sample depth value at the sample points $\boldsymbol{\omega} = (\omega_1, \ldots, \omega_n)$ to get a value that better corresponds to the depth values at the surrounding regions. We do this by multiplying the second term in Equation (4.4) by $\frac{1}{4}$, and denote this as the adjusted sample simplicial depth. We then get the adjusted depth of

$$SD_{\text{sia}}(\omega_a; \boldsymbol{\omega}) = \frac{1}{\binom{n}{3}} \sum_{\substack{1 \le i < j < k \le n \\ i,j,k \neq a}} \mathrm{I}\left(\omega_a \in \Delta(\omega_i, \omega_j, \omega_k)\right) + \frac{3}{4n}, \tag{4.5}$$

at sample points ω_a. If x is not in $\boldsymbol{\omega} = (\omega_1, \ldots, \omega_n)$, the adjusted sample simplicial depth is the same as the sample simplicial depth. If we now return to Figure 4.7, we can see the effect of this. The values printed in green are the values for the adjusted sample simplicial depth at the five sample points. Comparing these with the values for the sample simplicial depths, printed in purple, we see that the new values corresponds better to the depth values in the regions surrounding the points. This is further illustrated in Figure 4.9, where an approximate 75% probability region is drawn for a data sample of size 99 (the details of how these regions are calculated are described later in Section 4.4). The 74 sample points that have a depth higher than the cut-off value is plotted as a purple circle, while the 25 sample points that have a lower depth is plotted as green triangles. The probability region constructed using the simplicial depth clearly shows that many of the sample points with a depth higher than the cut-off value are located outside the region. This means that the neighbourhood of these points have depth-values below the cut-off point, so the simplicial depth value at the sample point doesn't really represent well the depth values in the neighbourhood. Using the adjusted simplicial depth, as shown in the plot to the right, generates a probability region that better corresponds to the depth values at the sample points. Note also that the region is larger in the case of the adjusted simplicial depth, simply because the cut-off value will be a constant lower than in the case of the simplicial depth.

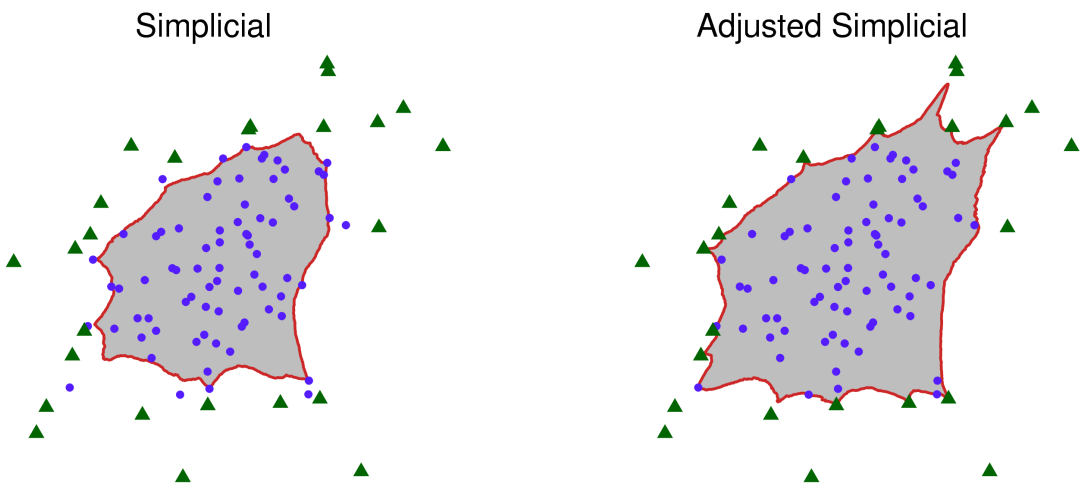

Figure 4.9: Two approximate 75% probability regions constructed for a data sample of size 99, using the simplicial depth (left) and the adjusted simplicial depth (right). The purple circles shows the sample points which have a depth value above (or equal) to the cut-off value, while the green triangles show the points which have a depth value lower than the cut-off value.

4.3.4 Tukey's Depth

Tukey's depth was introduced in (Tukey, 1975). If we restrict ourself to $\mathbb{R}^2$, we got the following formula for the depth (Liu and Singh, 1997):

$$D_{\text{tuk}}(x;F) = \inf_E \{P(E) : \text{E is a closed half-space in } \mathbb{R}^2 \text{and } x \in E\},$$

where $I(\cdot)$ is the indicator function. The sample Tukey's depth is given by

$$SD_{\text{tuk}}(x;\boldsymbol{\omega}) = \inf_E \left\{ \left(\frac{1}{n} \sum_{i=1}^{n} I(\omega_i \in E) \right) : \text{E is a closed half-space in } \mathbb{R}^2 \text{and } x \in E \right\}, \tag{4.6}$$

Calculation of the depth at a point x boils down to drawing a straight line through x and counting the number of points that lie on one side of the line. The depth is then the minimum value obtained when this is done for all possible lines through x. The sample depth is illustrated in Figure 4.10, where the two data sets are the same as in Figure 4.7.

The picture will of course be much more complicated when n is large. This can be seen in Figure 4.11, where the sample Tukey's depth is plotted in R for a sample of size $n = 35$ from the bivariate

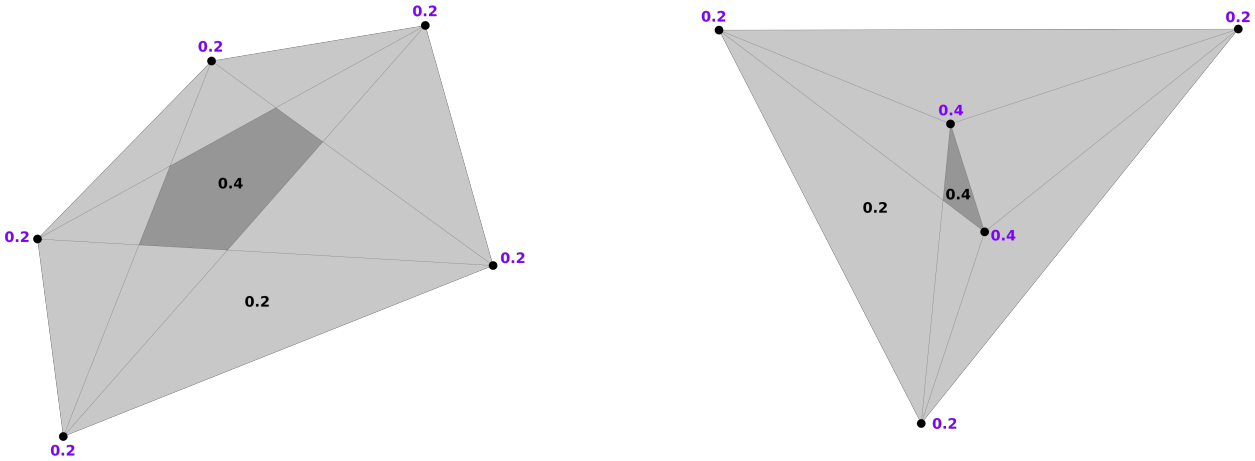

Figure 4.10: Illustration of the sample Tukey's depth for two data sets of size 5. More shading corresponds to a higher depth. The depth value at the sample points are printed in purple.

normal distribution, with parameters given by Equation (4.1). From this plot we can notice two things

- There are a finite number of possible depth values, which follows from the fact that the sample Tukey's depth works by counting points. The increase in depth value between each jump is equal to $\frac{1}{n}$, because one more point is counted in Equation (4.6).
- All the boundaries between these regions of different depth values consists of straight lines. Actually, if one studies Figure 4.11 closely, one can notice that all these lines actually are a subset of all possible straight lines between the sample points (not necessarily starting or ending at sample points).

4.3.5 Angle Depth

Here we try to introduce a new function for calculating a kind of sample depth of a points x with respect to a data cloud $\boldsymbol{\omega} = (\omega_1, \ldots, \omega_n)$ in $\mathbb{R}^2$. The idea is to look at all angles formed by drawing straight lines from x to all the points in $\boldsymbol{\omega}$, and finding the maximum value $\theta_{\max}$ of all the n angles between successive lines. The formula for the sample depth is then

$$SD_{\text{ang}}(x; \boldsymbol{\omega}) = 1 - \frac{\theta_{\max}}{2\pi}.$$

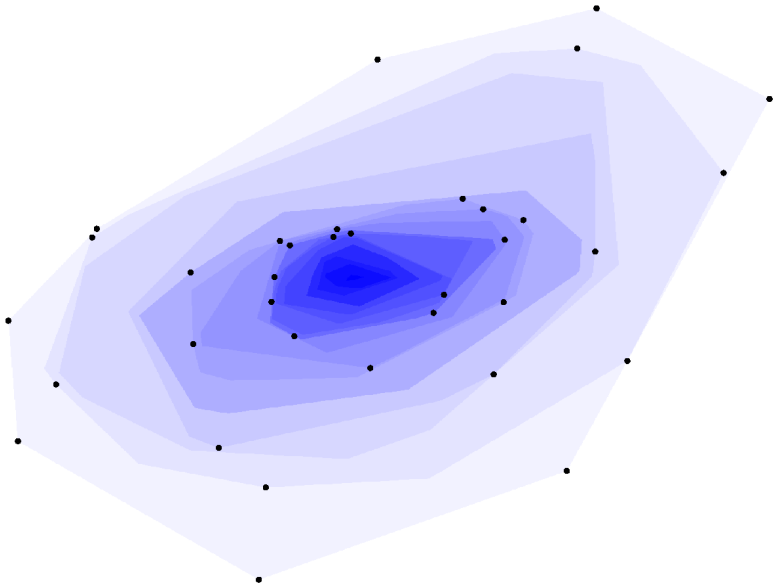

Figure 4.11: Illustration of the sample Tukey's depth, using data of size $n = 35$ generated from the bivariate normal distribution. A dark blue color corresponds to a high depth.

We denote this as the sample angle depth, even though we have not defined a corresponding depth function $D_{\text{ang}}(x; F)$. Figure 4.12 illustrates how the value of $\theta_{\max}$ is found for a sample of size four, where the four angles are drawn as arcs. The largest angle, $\theta_{\max}$, will here be the one between ω_1 and ω_3.

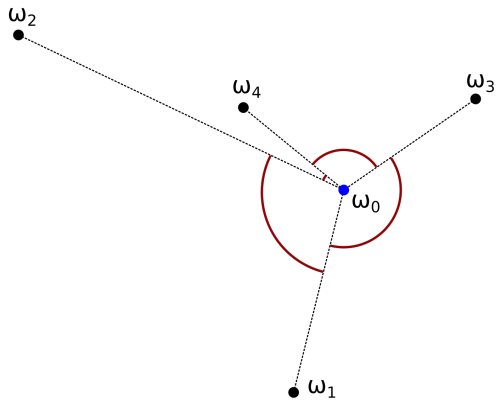

Figure 4.12: Illustration of how the sample angle depth is calculated.

The idea is that points deep inside the data cloud will have a small value of $\theta_{\max}$, because there are other sample points in all directions around it. A point that lies on the outskirts of the data cloud will have a big value for $\theta_{\max}$, because they all lie in pretty much the same direction from

this sample point. Figure 4.13 illustrates how the sample angle depth looks for random samples of size $n = 5$ (left) and $n = 15$ (right) from the bivariate normal distribution. Note that these are the same data samples that were used to illustrate the sample Mahalanobis depth in Figure 4.5 and the sample simplicial depth in Figure 4.6. There are several differences between these three figures, but one thing to notice is that the sample angle depth obtains quite a large value in the area between the three points furthest to the left in the case of $n = 15$, compared to the two other sample depths.

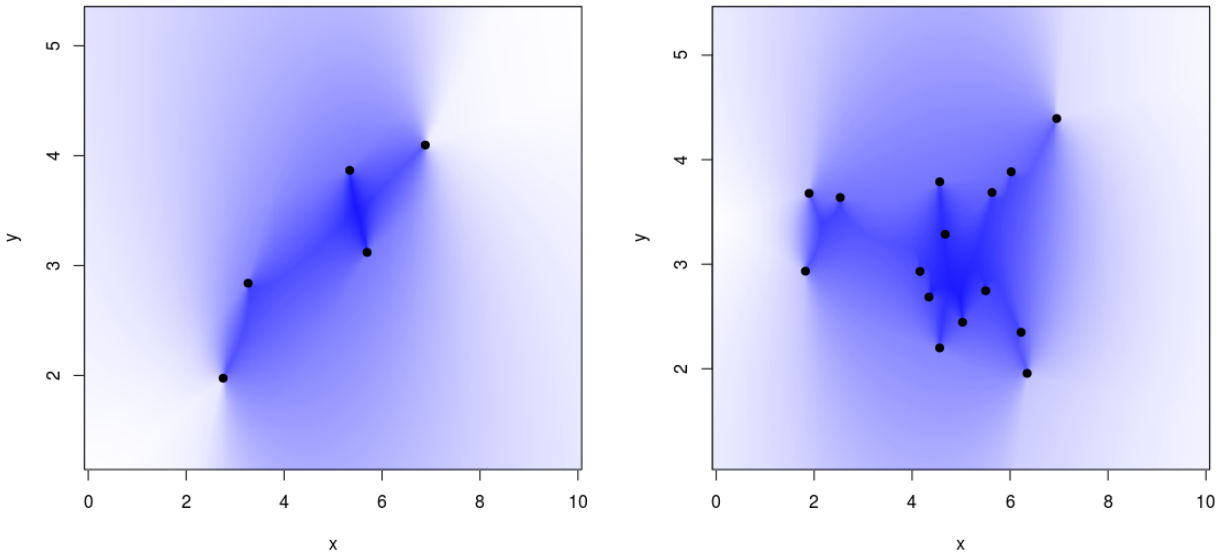

Figure 4.13: Illustration of the sample angle depth, using data generated from the bivariate normal distribution, with sample sizes $n = 5$ (left) and $n = 15$ (right). A dark color corresponds to a larger value of the depth.

4.4 Probability Region from Depth

Now it's time to use the sample depths to construct approximate probability regions. The goal is to use a random sample of size n from some distribution F to construct a region that has a certain probability of containing a new generated value from F. Here we only want to discard the points that are far away from the data cloud, and so it makes sense to construct a one-sided probability interval where only the points with the lowest depth values are discarded. After these points are removed, one needs to translate the boundary of the probability interval from the

one-dimensional point at some value $SD(x;\boldsymbol{\omega}) = a$, from now on called the cut-off value, to a two-dimensional boundary curve. The probability region will be all points inside this curve, and its shape and area will depend on what coverage probability level are chosen and which depth function is used.

To calculate the boundary of the probability region in R, we calculate the sample depth value at each point on a large grid, and then use the contourLines-function from the grDevices-library. This function returns the vertices of the contour line. The grids used in the plots later are 1000x1000 in size.

In general, the algorithm for calculating the probability regions are as follows:

Algorithm 3 - Probability regions from data depth

Input: A random sample $(\mathbf{x},\mathbf{y})$, coverage probability level $(1-\gamma)$ and number of simulations m.

1: Calculate the sample depth for each point in the data cloud $(\mathbf{x},\mathbf{y})$.

2: Sort the data depths and find the cut-off value for the sample depth corresponding to the desired coverage probability level of the interval.

3: Define a large regular grid over a region containing at least all the sample points, and calculate the sample depth at each of these points.

4: Use the contourLines-function from the grDevices-library in R to calculate the boundary of the probability region, using the sample depth values over the whole grid and the calculated cut-off value. The probability region will be all points inside this boundary.

Note that steps 3 and 4 will be a bit different when using the Tukey's depth, and also for the case when a convex hull is constructed instead of a contour line. This will be explained below.

Now we discuss how the different sample depths are calculated in R, and the shapes of the various probability regions.

4.4.1 Mahalanobis Depth

If the sample depth function is the Mahalanobis depth, then the boundary will always be an ellipse. This is because the boundary satisfies

$$\left[1 + (x - \bar{\boldsymbol{\omega}})^T \hat{\Sigma}_{\boldsymbol{\omega}}^{-1} (x - \bar{\boldsymbol{\omega}})\right]^{-1} = a, \tag{4.7}$$

which can be interpreted as the contour $g(x) = \frac{1-a}{a}$ of the quadratic form

$$g(x) = (x - \bar{\boldsymbol{\omega}})^T \hat{\Sigma}_{\boldsymbol{\omega}}^{-1} (x - \bar{\boldsymbol{\omega}}).$$

This has the same form as the exponent in the kernel $e^{\frac{-1}{2}(x-\mu)^T \Sigma^{-1}(x-\mu)}$ of the bivariate normal density $X \sim \mathrm{N}_2(\mu, \Sigma)$, which we know have elliptic contours for any covariance matrix Σ. Hence, the contours defined by Equation (4.7) are also ellipses for any empirical covariance matrix $\hat{\Sigma}_{\boldsymbol{\omega}}$.

One can see from Figure 4.5 that the contour lines indeed will be ellipses. The calculation of the Mahalanobis depth is implemented directly in R using Equation (4.2).

4.4.2 Simplicial and Adjusted Simplicial Depth

In general, the boundary curve will have a much more complicated shape than an ellipse. For instance, the sample simplicial depth (and the adjusted one) will have a boundary curve that is the union of many straight line segments. This is because the sample depth is defined in terms of triangles. See for instance Figure 4.6.

A function for calculating the sample simplicial depth was implemented in R. However, there exists a library called ”depth” in R containing a function with the same name, that can calculate a variety of sample depths, including the simplicial depth. The performance of these two functions was tested by calculating the sample depth at 100 sample points from a bivariate normal distribution, as well as on 100 new points using the 100 first sample points as the data cloud $\boldsymbol{\omega}$. Both functions gave the same sample depth at all 200 points, but the function from the depth-

library was much faster (0.025 seconds versus 0.131 seconds for all the 200 points). This function is based on (Rousseuw and Ruts, 1996), where they state that this algorithm runs in $\mathcal{O}(n \log n)$ time, while the function we have implemented here runs in $\mathcal{O}(n^3)$, because the number of triangles is approximately equal to $\frac{n^3}{6}$. Therefore, the function from the depth-library was used in the calculations for the sample simplicial depth in R. This was also used to calculate the adjusted simplicial depth, only that $\frac{3}{4}\frac{3}{n}$ were substracted from the value at sample points to get the adjusted values correct at these points, see Equation (4.5)

4.4.3 Convex Hull

An alternative to the generally complicated shapes of the probability regions, is to construct a convex hull of all $(1-\alpha)100\%$ sample points that have a depth more than or equal to the boundary value, as is done for the Tukey's depth in (Yeh and Singh, 1997). The convex hull of a set of points $\mathbf{y} = (y_1, \ldots, y_n)$ in $\mathbb{R}^2$ is the intersection of all sets in $\mathbb{R}^2$ that contains $\mathbf{y} = (y_1, \ldots, y_n)$ (Bærentzen et al., 2012). How this looks is illustrated in Figure 4.14, where the convex hull of two different data clouds are drawn. The boundary colored in orange is a union of straight line segments. It is clear that these regions look "simpler" than some of the more irregular shapes generated by the contour-function. It will also be much faster to calculate and plot, since there is no need to ever calculate the depth at locations other than the sample points, see

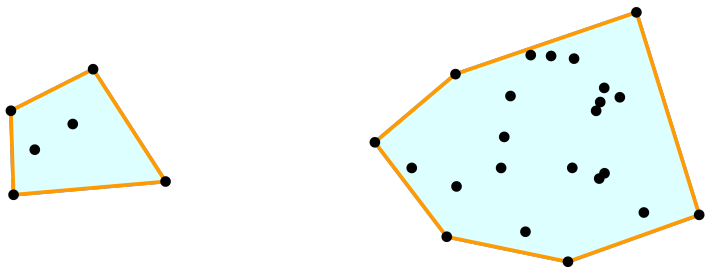

Figure 4.14: Convex hulls of two data clouds of different sizes. The orange lines shows the boundary of the convex hulls.

Here we choose to construct such a convex hull using the simplicial depth. Using the adjusted simplicial depth would give the same results, since the ordering of the sample points by depth-values are the same in both cases. This is because the adjusted simplicial depth subtracts the

same constant $\frac{3}{4}\frac{3}{n}$ from the simplicial depth values at each sample point, which can be seen from comparing Equation (4.4) with Equation (4.5).

4.4.4 Tukey's Depth

Because of the way Tukey's depth is calculated in $\mathbb{R}^2$, as explained in Section 4.3.4, we have that the boundary also in this case will be a union of straight line segments.

A function in R was made that calculates the sample Tukey's depth, but this implementation was very slow. Just as in the case of the simplicial depth, we used the function from the depth-library in R to calculate the sample depths. The comparison of these functions on the same 200 points used for the simplicial depth, we obtained that all the 200 depth values for the Tukey's depth were equal, but that the function from the depth-library was much faster (0.024 seconds versus over 2 minutes). The algorithm in this function is based on (Rousseuw and Ruts, 1996), the same paper as for the simplicial depth. The library also have a function called "isodepth" which was used to compute the vertices of the contour line, instead of using a 1000x1000-sized grid as described earlier.

4.4.5 Angle Depth

From the illustration in Figure 4.13 we would expect that the sample angle depth will give confidence regions with quite erratic boundaries. This is also the case, as we will see later.

The function for calculating the sample angle depth is implemented in R, and calculates the value by the following simple algorithm:

Algorithm 4 - Sample angle depth

Input: A data cloud $\boldsymbol{\omega} = (\omega_1, \ldots, \omega_n)$ and a point $x \in \mathbb{R}^2$.

1: Calculate the angles $\theta = (\theta_1, \ldots, \theta_n)$ for each of the straight lines connecting x with the n sample points (as in polar coordinates). Note that if x is one of the sample points in $\boldsymbol{\omega}$, then x is removed from the data cloud and we only look at the angles to the $n-1$ other sample points.

2: Sort these angles, obtaining the sorted list $\left(\theta_{(1)}, \ldots, \theta_{(n)}\right)$.

3: Calculate a vector of length n containing the difference between each consecutive angle in this the sorted list, including the difference $\theta_{(1)} - \theta_{(n)}$.

4: Find $\theta_{\max}$ as the maximum value of these n angle differences (or $n-1$ angle differences, if x is one of the sample points), and return $SD_{\text{ang}}(x; \boldsymbol{\omega}) = 1 - \frac{\theta_{\max}}{2\pi}$.

4.5 Plots of Probability Regions

What defines a good confidence region? If we look at confidence intervals, we generally want them to be exact, meaning that the nominal coverage is equal to the true coverage, and that the width of the interval is as small as possible. Similarly, when using confidence regions we want them to be exact and to have as small area as possible. It's also preferable if the region have a simple shape, like a convex hull. We can say the same things about probability regions and intervals.

The two data samples used to test the various probability regions are shown in Figure 4.15. One data set is from the bivariate normal distribution, with parameters given by Equation (4.1), while the other is from the UnitCircle-distribution, as defined earlier. Both samples are here of size $n = 999$.

In Figure 4.16 the probability regions are plotted for different choices of the depth function, using the data sample of size $n = 999$ from the bivariate normal distribution. The coverage probability level was chosen to be $(1-\gamma) = 0.95$, and the grid used to calculate the boundary is 1000x1000 in size. The area of each probability region is printed in the plots to the left. All

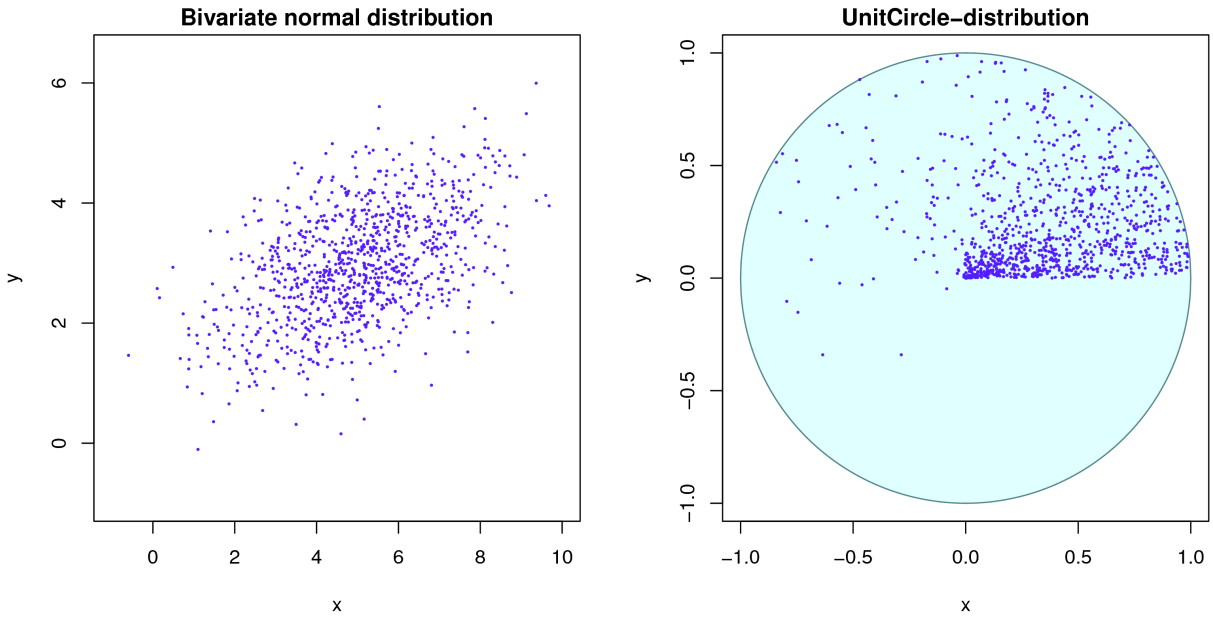

Figure 4.15: Two data samples of size $n = 999$, one from the bivariate normal distribution (left), and one from the UnitCircle-distribution (right).

the plots to the right are zoomed in versions of the plots to the left, where the portion of the plot that are shown are indicated by a dotted square in the plot to the left. Similar plots for the UnitCircle-density are shown in Figure 4.17.

From the plots of the various probability regions we notice the following

- When the data is from the bivariate normal distribution, we see that the regions look quite similar, except for the smoothness of the boundary. We see that Mahalanobis, simplicial convex hull and Tukey's have quite simple boundaries, while the three other regions have much more erratic boundaries. This is also the case when the data is generated from the UnitCircle-distribution.

- The probability region generated using the adjusted simplicial depth in Figure 4.17 looks a bit strange. Actually, the probability region stretches out to all the $n = 999$ sample points, but it does not include these "outer" points. The same happens with the angle depth for this underlying distribution, and almost also for the angle depth in Figure 4.16, although four or five points are clearly separated from the region.

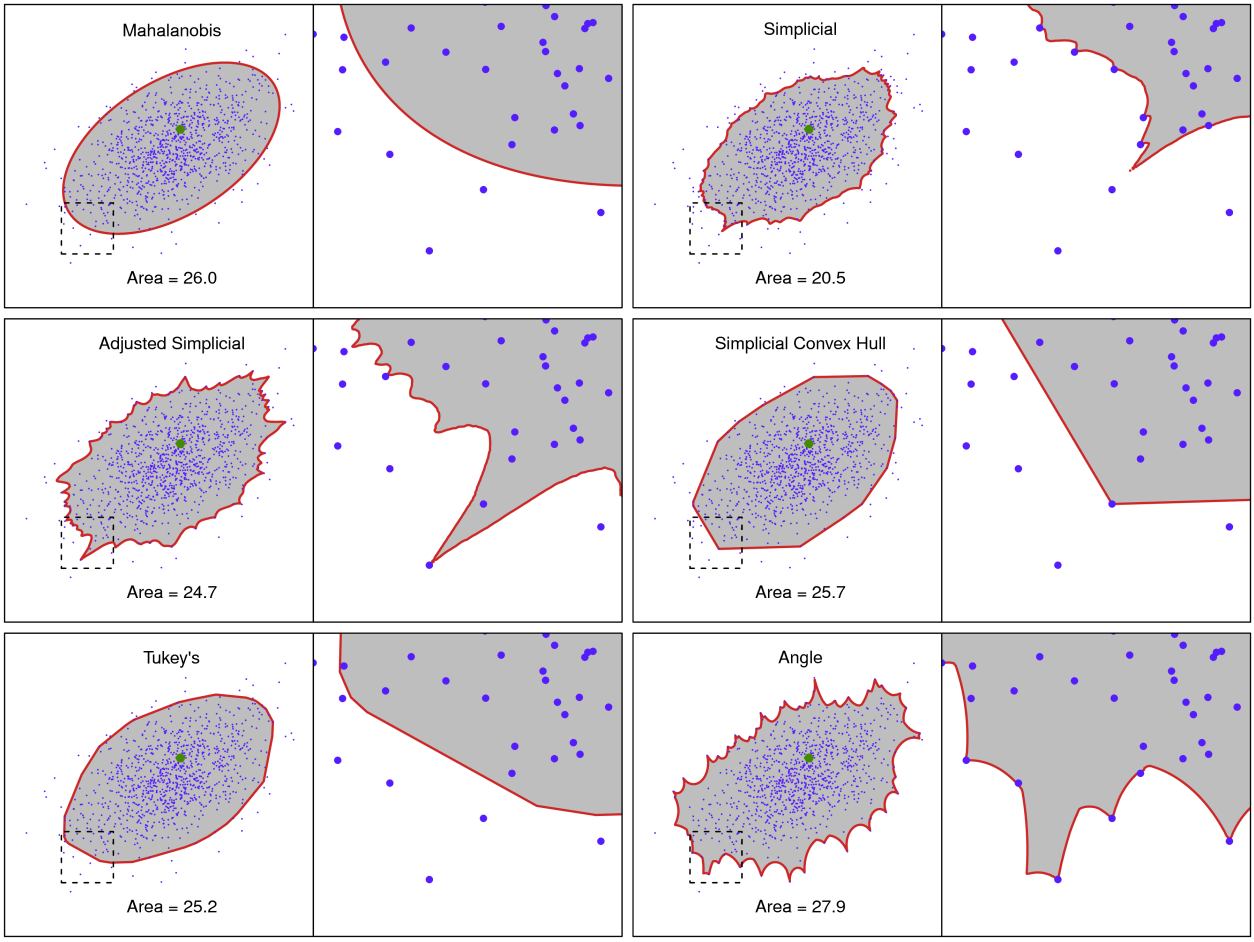

Figure 4.16: Probability regions drawn for the different types of depth functions, using the data sample from the bivariate normal distribution. The depth type is printed above the region, while the corresponding area is printed below. The plots to the right shows a close-up version of the regions, to get a better view of the shape of the boundary. The part of the probability region that is zoomed in on is shown as a dotted box in the plots to the left. The green dot shows the location of a new generation from the same distribution.

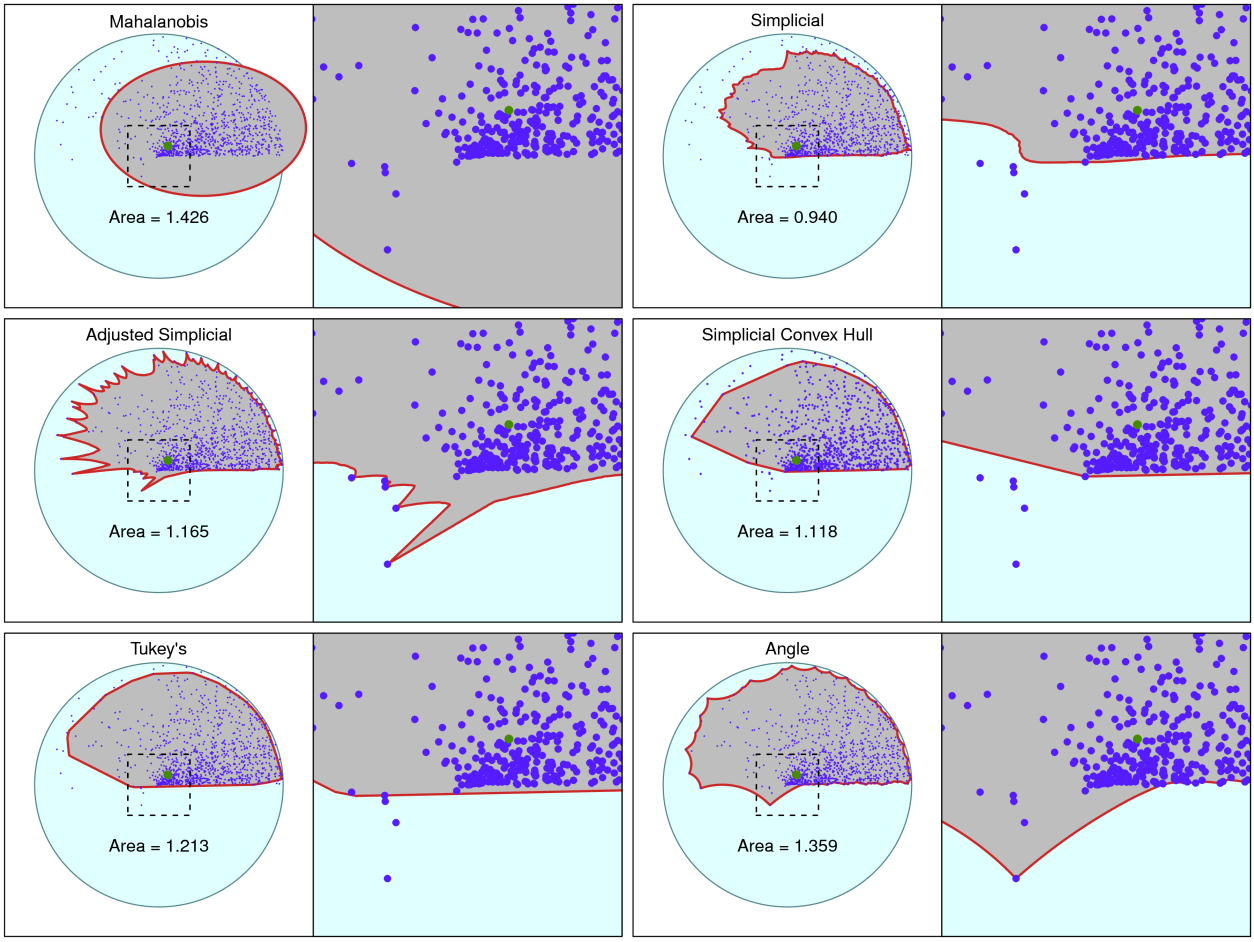

Figure 4.17: Probability regions drawn for the different types of depth functions, using the data sample from the UnitCircle-distribution. The depth type is printed above the region, while the corresponding area is printed below. The plots to the right shows a close-up version of the regions, to get a better view of the shape of the boundary. The part of the probability region that is zoomed in on is shown as a dotted box in the plots to the left. The green dot shows the location of a new generation from the same distribution.

- With data from the normal distribution we observe that the area of the region is smallest for the simplicial depth and largest for the angle depth. When the data is from the UnitCircle-distribution we still have that the simplicial depth constructs a region with the smallest area, but now the largest area belongs to the region from the Mahalanobis depth. The reason for this is that Mahalanobis always creates an ellipse as the boundary, and this is not a suitable shape for this type of underlying distribution. Clearly, much of this ellipse is covering regions where there are no points at all.

4.6 Coverage of Probability Regions from Depth Statistics

Here we calculate the coverage proportion for each type of depth function, using a coverage probability level of 0.95. This is done by generating a random sample of size n from either the bivariate normal distribution, with parameters given by Equation (4.1), or the UnitCircle-distribution. This is done a lot of times, up to $2^{11} = 2\ 048$ iterations where a new sample is generated each time. At each iteration we calculate all the n sample depths, and check if the sample depth of a new generation from the same distribution is inside the probability region. This is the same as checking if the sample depth of this point is above or equal to the cut-off value, meaning we never actually have to compute all the vertices in the boundary of the probability region. This saves a lot of computation time, because we do need to calculate the sample depths at all the points in a 1000x1000-grid. The exception to this is of course the probability region constructed using a convex hull, where we actually have to check if the point is inside this polygon region. This is done using a function called pnt.in.poly from the SDMTools-library in R, which simply checks if a point is inside a polygon defined by its vertices.

We do this for different sample sizes n to see get an idea of how many sample points is needed to get an approximately correct coverage proportion of 0.95. The results of this is shown in Figures 4.18 and 4.19, with the same vertical lines as in the plots in the previous chapter, showing approximate 95% probability intervals for the coverage proportion, under the assumption that the true coverage probability is 95%. Here the logarithm used is in base 2. From these figures we note the following for the six different methods for constructing the regions:

- Using the Mahalanobis depth on the normal distribution seems to give correct coverage even for small n. When the data is from the UnitCircle-distribution, however, we have that the coverage proportion is a bit too low, and actually it doesn't seem to improve significantly when the sample size n is increased. As noted before, this is likely because of the fact that the Mahalanobis depth constructs ellipses, which doesn't work well with the asymmetric UnitCircle-distribution.

- The calculation of coverage proportions using both the simplicial depth and the adjusted simplicial depth gives similar results. Notice, however, that the coverage proportion is closer to 0.95 for the adjusted one when using both $n = 99$ and $n = 999$ for both types of underlying distributions (the plot from the simplicial depth have a different scale than the rest).

- There seems to be a problem with the regions constructed using simplicial depth and a convex hull, namely that the coverage proportion actually doesn't stop at 95% when n is increased, but continuous to rise past this value. This is of course not what we want, and the reason might be the following: When n is large we have that the proportion of points that lies furthest out (for instance all the vertices in the convex hull constructed from all the sample points), is so large that the cut-off value actually is the sample depth values at these points (or close to it). This was probably what we observed in the plot of the adjusted simplicial region in Figure 4.17, where the cut-off value clearly is the same as the values for the depth at the outer sample points. We are unsure of how this can be fixed to get coverage proportions closer to 95%, it this is in fact possible using a convex hull.

- The Tukey's depth gives coverage proportions that seems to approach 95% fairly fast as m increases, altough a maybe bit slower in the case of the UnitCircle-distribution.

- The angle depth actually look really good, for some reason it seems that the coverage proportions are close to 95% already at $n = 99$ for both types of underlying distributions used here. Notice that the sample sizes n are smaller for the angle depth than for the rest in Figure 4.18 and 4.19, which is simply because it takes such a long time to compute.

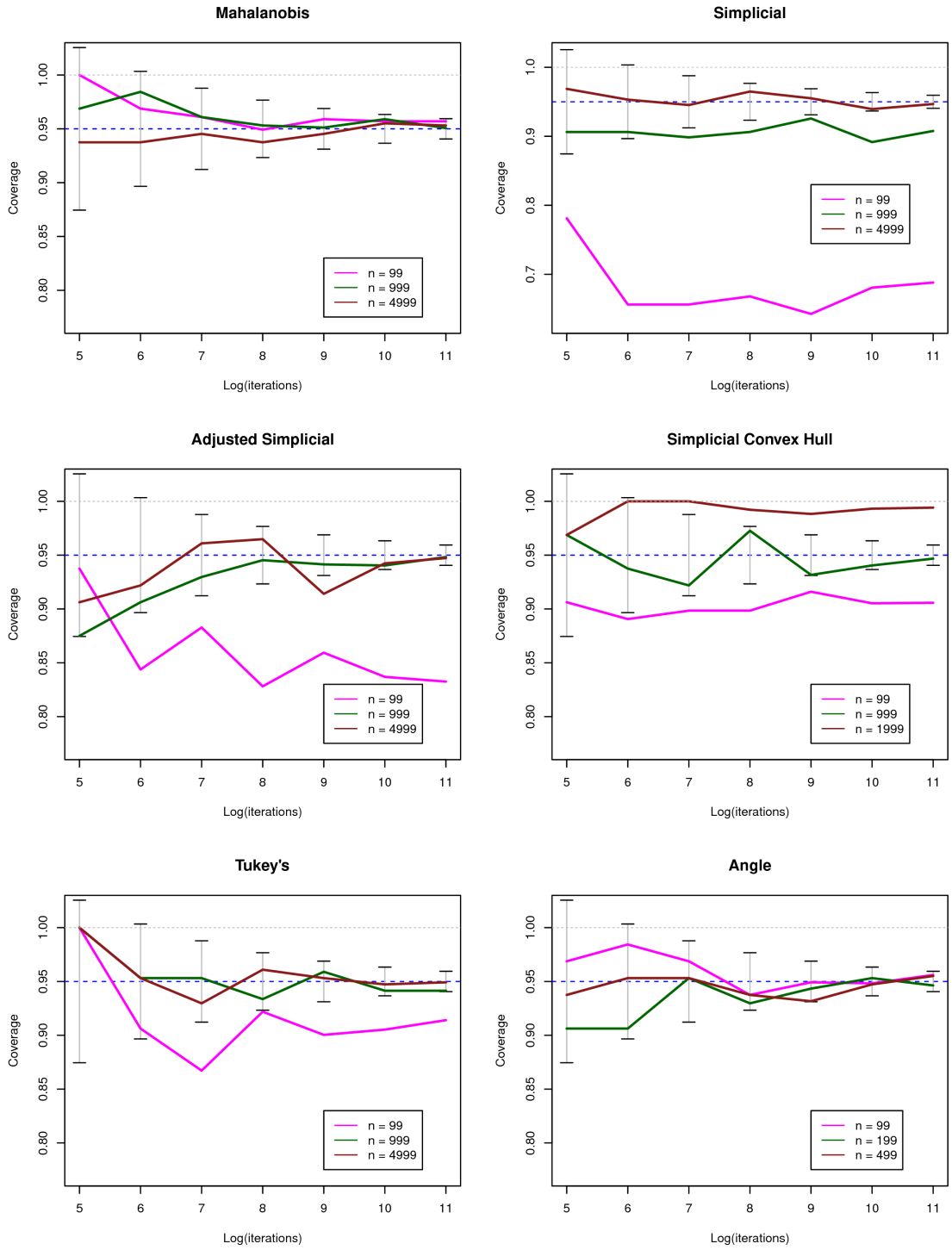

Figure 4.18: Convergence proportion as a function of the logarithm (with base 2) of the number of iterations, for probability regions using data from the bivariate normal distribution. Three different values for the sample size n is used in each plot, and the coverage probability level is $(1-\gamma) = 0.95$ in all cases. The vertical lines are the same approximated probability intervals as used in earlier plots. Note that the scale on the y-axis is different for the simplicial depth.

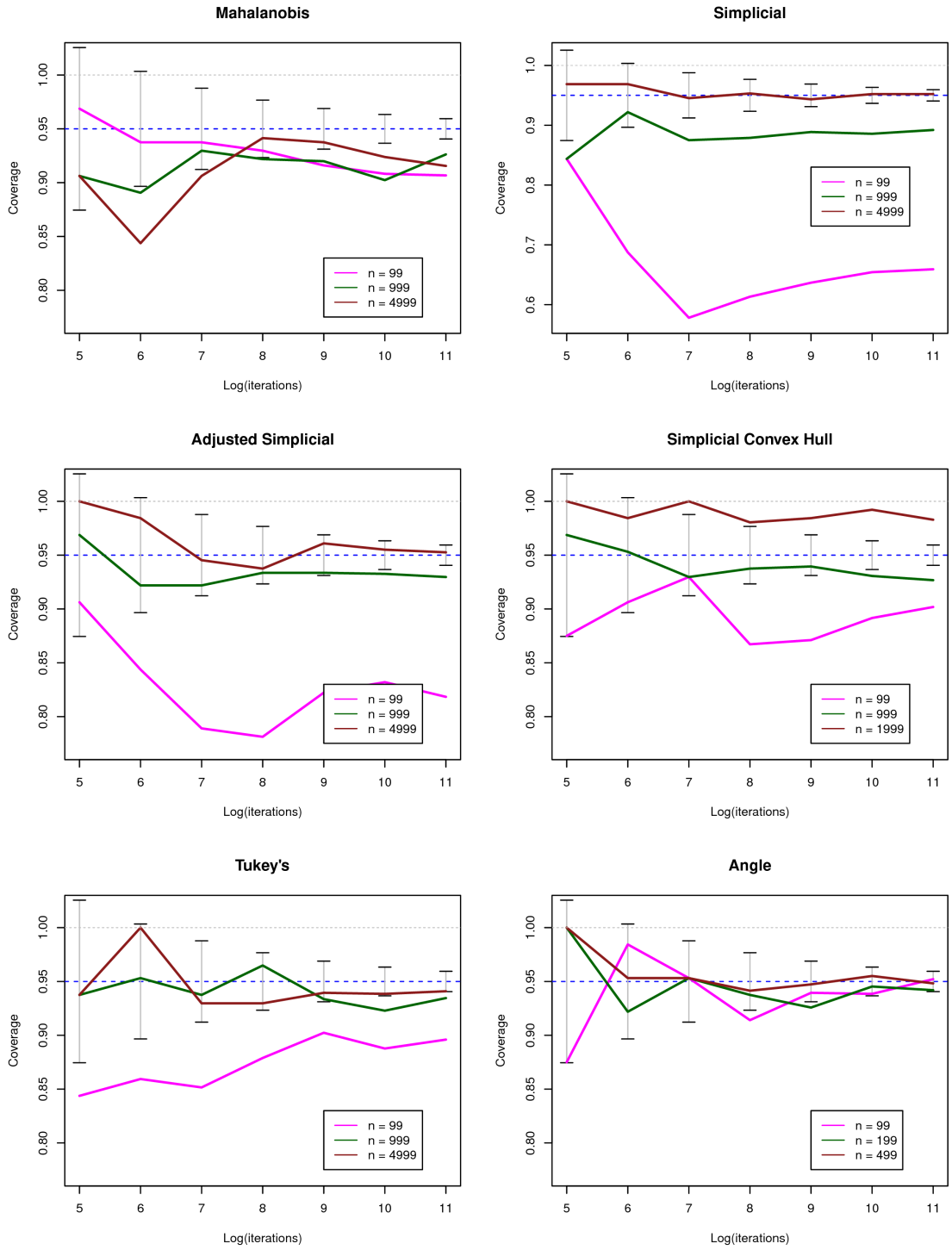

Figure 4.19: Convergence proportion as a function of the logarithm (with base 2) of the number of iterations, for probability regions using data from the UnitCircle-distribution. Three different values for the sample size n is used in each plot, and the coverage probability level is $(1-\gamma) = 0.95$ in all cases. The vertical lines are the same approximated probability intervals as used in earlier plots. Note that the scale on the y-axis is different for the simplicial depth.

Chapter 5

Confidence Regions for the Gamma Distribution

In this chapter we apply the methods of the previous chapter to generate approximate confidence regions the parameters in the gamma distribution. Values for $\tilde{\alpha}$ and $\tilde{\beta}$ are generated using the same steps as in Algorithm 2 in Chapter 3, but here we look at the distribution of these values $(\tilde{\alpha}, \tilde{\beta})$ in $\mathbb{R}^2$. It has been shown in (Lindqvist and Taraldsen) that the distribution of $(\tilde{\alpha}, \tilde{\beta})$ is an exact two-dimensional confidence distribution of the true parameters (α, β). We look at four different methods for constucting the approximate confidence regions:

- Using the Mahalanobis depth
- Using the adjusted simplicial depth
- Using the simplicial depth and constructing a convex hull.
- Using the Tukey's depth.

The whole algorithm for this, including the steps to generate values for $\tilde{\alpha}$ and $\tilde{\beta}$, are shown below.

Algorithm 5 - Confidence regions for the gamma distribution

Input: A random sample $\mathbf{x}$, confidence level $(1-\gamma)$ and number of simulations m.

1: Calculate $T_1(\mathbf{x})$ and $T_2(\mathbf{x})$ using Equations (3.6) and (3.7).

Iterate m times:

2: Generate $\mathbf{U}^* = (U_1^*, \ldots, U_n^*)$, where each U_i^* is drawn independently from the uniform distribution between 0 and 1.

3: Solve $T_2(\mathbf{x}) = g_2(\mathbf{u}^*, \tilde{\alpha})$ numerically for $\tilde{\alpha}$ using Equations (3.7) and (3.10). This is done using the ... function in R.

4: Solve $T_1(\mathbf{x}) = g_1(\mathbf{u}^*, \tilde{\alpha}, \tilde{\beta})$ for $\tilde{\beta}$ using Equations (3.6) and (3.9), after inserting the solution for $\tilde{\alpha}$ in the previous step.

End iteration

5: Calculate the sample depth for each point in the data cloud $\left(\tilde{\boldsymbol{\alpha}}, \tilde{\boldsymbol{\beta}}\right) = \left((\tilde{\alpha}_1, \tilde{\beta}_1), \ldots, (\tilde{\alpha}_m, \tilde{\beta}_m)\right)$.

6: Sort the data depths and find the cut-off value for the sample depth corresponding to some desired level of the confidence interval.

7: Define a large regular grid over a region containing at least all the sample points, and calculate the sample depth at each of these points.

8: Use the contourLines-function from the grDevices-library in R to calculate the boundary of the confidence region, using the sample depth values over the whole grid and the calculated cut-off value. The confidence region will be all points inside this boundary.

Note that steps 7 and 8 will be a bit different when using the Tukey's depth, and when a convex hull is constructed instead of a contour line. This was also the case in Algorithm 3 in Section 4.4, and now we use the same method for these two exceptions. The details of this can be found further down in Section 4.4.

Note that one is free to choose the number of simulations m in the above algorithm. Hence, if the true coverage probability of the confidence region approaches the nominal coverage probability as $m \to \infty$, one can choose m high enough to get approximately the desired coverage

probability. We saw in Figure 4.18 and 4.19 that the coverage proportion for the most part got closer to 95% as the sample size increased, and we will investige the same for the confidence regions of (α, β) later in this chapter.

5.1 Results

The data sample we use in the plots later is

$$\mathbf{x} = [3.0, 3.1, 2.7, 4.5, 4.1, 9.2, 3.3]\,. \tag{5.1}$$

If we assume that the underlying distribution is the gamma distribution, we can compute the maximum likelihood estimates for the parameters using the fitdistr-function from the MASS-library in R. We then get the following estimates:

$$\hat{\alpha} = 5.88$$
$$\hat{\beta} = 1.38$$

We would expect these values to be inside the confidence regions constructed later, and this is also the case.

Figure 5.1 illustrates the distribution of $(\tilde{\alpha}, \tilde{\beta})$ obtained after the first 4 steps in Algorithm 5 above, using the data set defined in Equation (5.1). Here 10 000 generations of $(\tilde{\alpha}, \tilde{\beta})$ are plotted, and this gives an idea of what the underlying density function looks like.

The four different types of confidence regions generated using the chosen depths or methods listed above, are shown in Figure 5.2 and 5.3. Here each plot shows three confidence regions generated for different confidence levels: 50% (green boundary), 75% (yellow boundary) and 95% (orange boundary). The number of generated sample points for $(\tilde{\alpha}, \tilde{\beta})$ is 999, and the same data cloud is used in each plot, so we can compare the shapes directly. A small part of each confidence region is zoomed in and plotted in the upper left corner, to get closer look at how the boundary behaves. The areas of the three regions are printed in the legend.

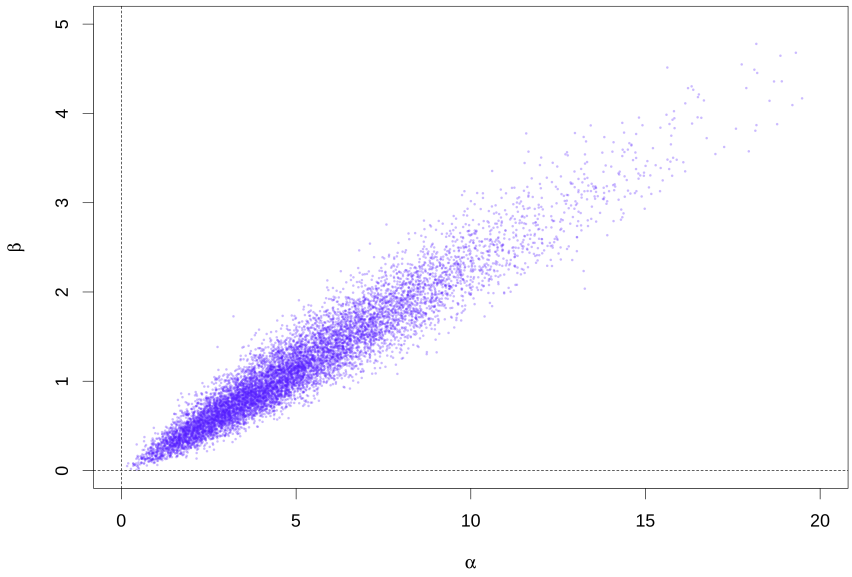

Figure 5.1: Plot of 10 000 generated values for $(\tilde{\alpha}, \tilde{\beta})$.

From Figure 5.2 and 5.3 we note the following

- The Mahalanobis depth generates an elliptical boundary, as it should. We observe that the 95%-confidence region extends far into the negative numbers in the bottom left corner. The parameter values of α and β are always positive, so it is not ideal to have such a large portion covering a region that have zero probability of containing the true parameter values. Because of this, it seems likely that the true coverage probability is below 95% in this case, as will be investigated later.

- The adjusted simplicial depth generates a confidence region that better mimicks the underlying distribution of $(\tilde{\alpha}, \tilde{\beta})$, at least there is no part that extends to negative values for α and β. The boundary is quite erratic, at least in the upper right and bottom left corners. In the zoomed in picture one can notice a couple of orange points from the boundary that lies a small distance outside the region, but this is probably just because of some numerical inaccuracies from calculating the border as a contour of function values calculated over a large grid.

- The convex hull constructed using the simplicial depth creates confidence regions that have quite simple shapes. One can say that it looks similar to the one from the adjusted

simplicial depth, only now the boundary have become much less erratic.

- The confidence region generated using the Tukey's depth looks very similiar to the previous one using a convex hull, only that there seems to be more vertices in the boundary, see for instance the part that is zoomed in on.

There is not a huge difference in values for the area of the confidence regions generated using the four different approaches. If we restrict ourselves to the ones with confidence level of 95%, we see that the adjusted simplicial and the simplicial convex hull almost have the same area (10.9 and 11.0, respectively), which makes sense since the reach or extension of these regions should be quite similar. The Mahalanobis depth gives an area of 11.5, while the Tukey's depth constructs a region with the biggest area of 11.8.

5.2 Logarithmic Transformation

An alternative to constructing confidence regions directly from the data, is to use some transformation on the sample points. This can be useful if the transformed sample points seems to have a more "well behaved" distribution, for instance if the cloud have a shape similar to a random sample from the bivariate distribution. Here we use the logarithmic transformation, meaning that we calculate $\big(\ln(\mathbf{x}), \ln(\mathbf{y})\big)$ from the sample points $(\mathbf{x}, \mathbf{y})$, construct a confidence region in this new log-log-space, and then transform this region back to the original space. This will generally create a region that looks different from the one generated directly from the original data.

One thing to keep in mind is that the algorithm that generates the boundary of the confidence regions in R, returns a finite set of vertices, where the boundary is given by straight lines between these points. When transforming back from the log-log-space, straight lines will no longer be straight lines in the original space. An example of this is shown in Figure 5.4. Here the left plot shows a triangle in the log-log-space, while the right plot shows how this area looks after transforming back to the original space. The dotted lines in the plot to the right shows how the boundary would look if one just drew straight lines between the vertices. It is clear that the exponential transformation generally curves the lines between the vertices, and hence this

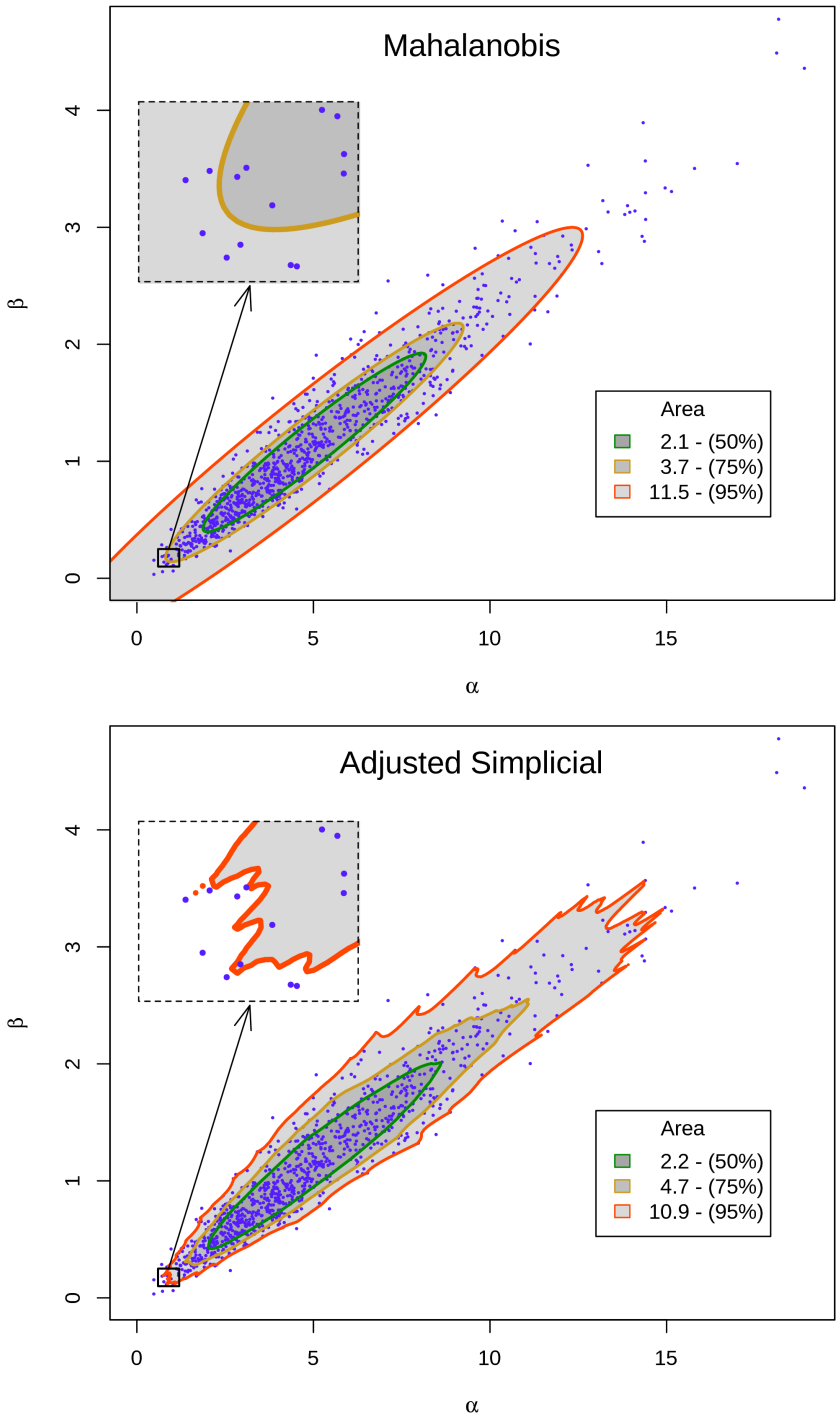

Figure 5.2: Plot of confidence regions constructed using the Mahalanobis depth (top) and the adjusted simplicial depth (bottom). Three different confidence levels are used: 95% (red border), 75% (yellow border) and 50% (green border). The corresponding areas of the confidence regions are listed in the legend. The rectangle in the upper left corner is a zoomed in picture of the small rectangle in the bottom left corner.

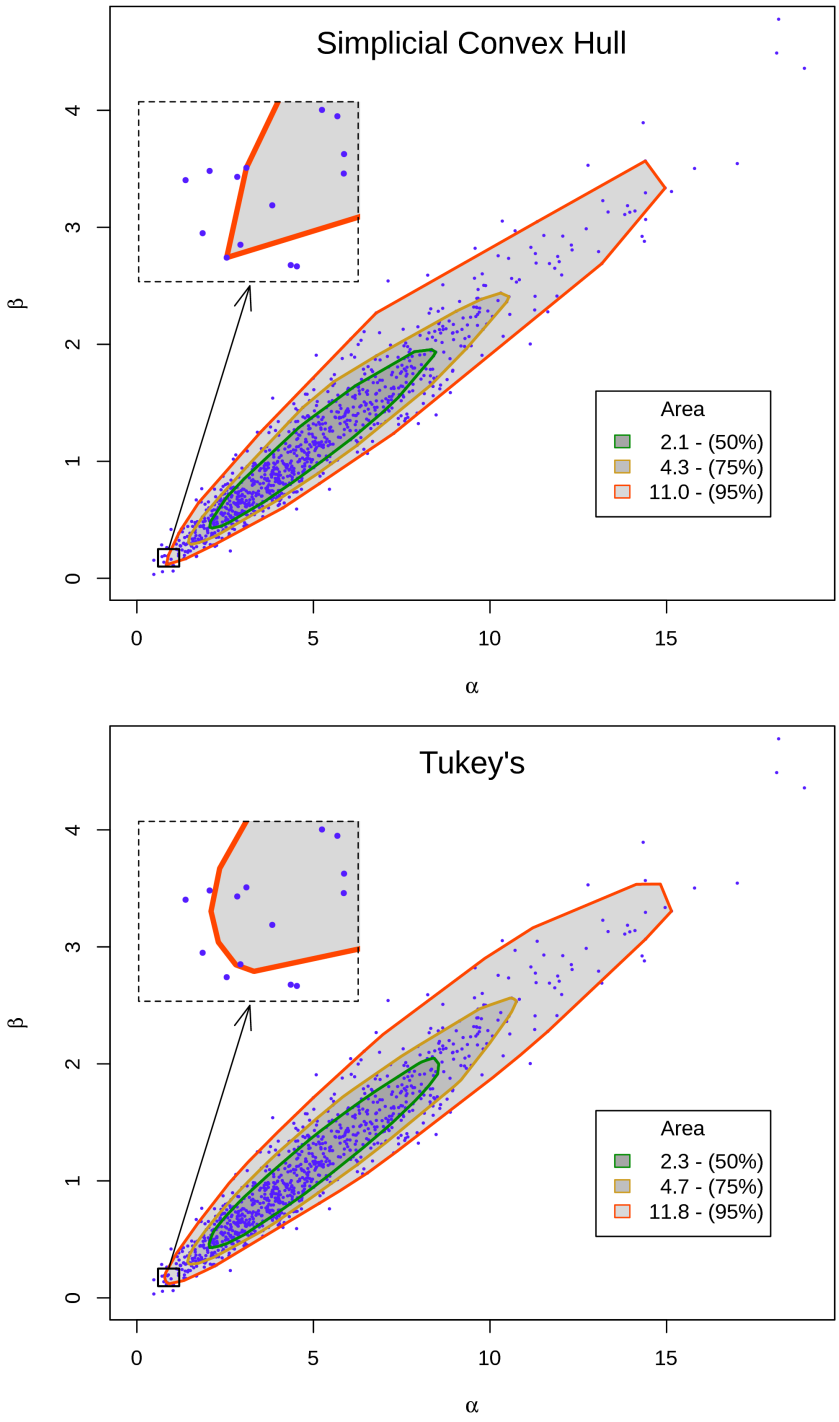

Figure 5.3: Plot of confidence regions constructed using the simplicial depth with convex hull (top) and the Tukey's depth (bottom). Three different confidence levels are used: 95% (red border), 75% (yellow border) and 50% (green border). The corresponding areas of the confidence regions are listed in the legend. The rectangle in the upper left corner is a zoomed in picture of the small rectangle in the bottom left corner.

must be accounted for when doing the back-transformation of the confidence region. This is especially important for the confidence regions constructed using a convex hull, since here the number of vertices in the boundary can be quite small and the line segments quite long.

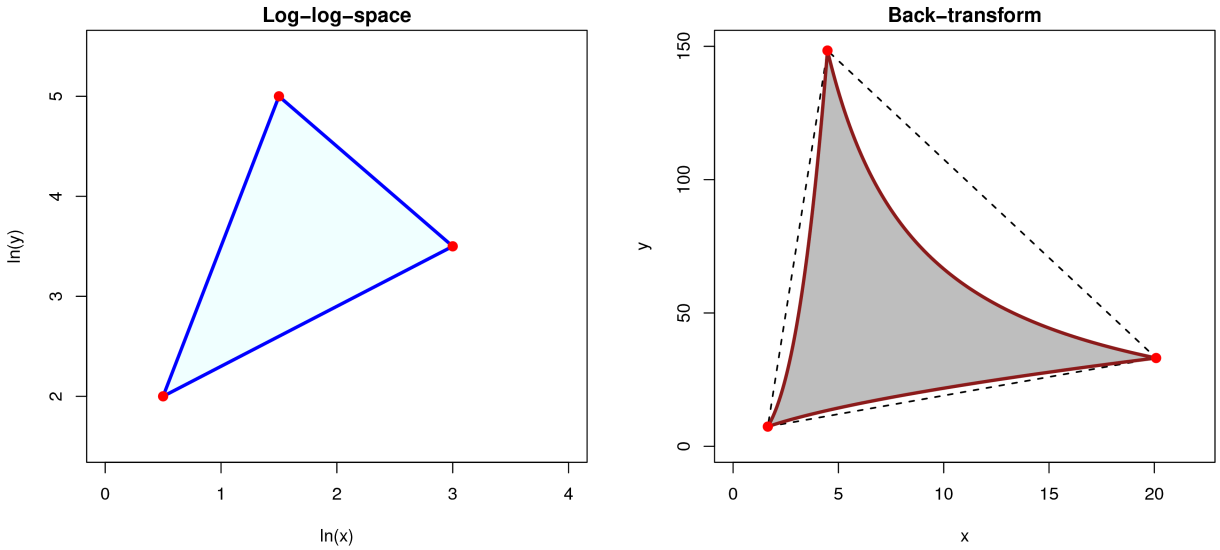

Figure 5.4: Illustration of how straight lines turns into curves under exponential transformation of both the coordinates. The left plot shows a triangle in the log-log-space, while the right plot shows how this region looks after using the exponential transformation on both coordinates x and y. The dotted lines shows straight lines between the vertices in the original space, for comparison.

The confidence regions constructed by using the log-log-transformation are shown in Figure 5.5 and 5.6.

- Clearly the Mahalanobis depth now gives a confidence region that is no longer an ellipse, but has a a shape more similar to a balloon. The areas are in this case significantly larger than when using the data sample directly. One thing to notice is that the confidence region no longer spans into negative values for α and β. This reason for this is of course that the exponential function is non-negative for all possible input values. We also observe that almost all sample points outside the 95%-confidence region now lies towards the bottom left corner, except one point in the upper right corner.

- The adjusted simplicial depth creates regions that have quite smooth boundaries, compared to the earlier plot from using the data directly. It is quite erratic in the bottom left corner though, as can be seen from the zoomed in window. Here we also see some orange

points that lie outside the rest of the confidence region, for instance a small patch in the upper right corner, but this is still probabily just some effects of the numerical calculations of the boundaries.

- Both the convex hulls constructed using the simplicial depth and the regions constructed using Tukey's depth, look very similar to the ones plotted earlier in Figure 5.3. One can observe that the vertices of the convex hulls after using logarithmic transformation is not exactly the same sample points as when using the original data directly. It is also possible to see that some of the lines now have some small curvature, for instance in the zoomed in window for the simplicial convex hull, although they are very close to straight lines.

The values for the area of the confidence regions are quite similar to the ones calculated earlier for the confidence regions using the original data directly. The exception is the Mahalanobis depth, which gets a much larger area.

5.3 Coverage of Confidence Regions

Here we investigate what coverage proportions we obtain using Algorithm 5 with the four different methods mention above for generating confidence regions. This is done by drawing a new sample of size $n = 20$ from the gamma distribution at each iteration, using true parameter values of $\alpha = 3$ and $\beta = 5$. For each new sample we generate m number of values for $(\tilde{\alpha}, \tilde{\beta})$, using steps 2-4 in Algorithm 5, before calculating sample depths and checking if the confidence region covers the true parameter values. This is done by calculating the depth of the true parameter value (α, β) with respect to the data cloud $(\tilde{\boldsymbol{\alpha}}, \tilde{\boldsymbol{\beta}})$, and comparing this sorted values of the depths calculated for all the points in $(\tilde{\boldsymbol{\alpha}}, \tilde{\boldsymbol{\beta}})$. The exception is for the simplicial convex hull, where compute the vertices of the boundary and check it the true parameter values lies inside this polygon, as we did in Section 4.6.

Algorithm 5 is quite slow compared to the earlier algorithms, because you first have to compute the cloud of points $(\tilde{\alpha}, \tilde{\beta})$, and then use these points to calculate depths and look at confidence regions. Hence, we test this algorithm for values of $m = 39$ and $m = 99$, where m is the number of

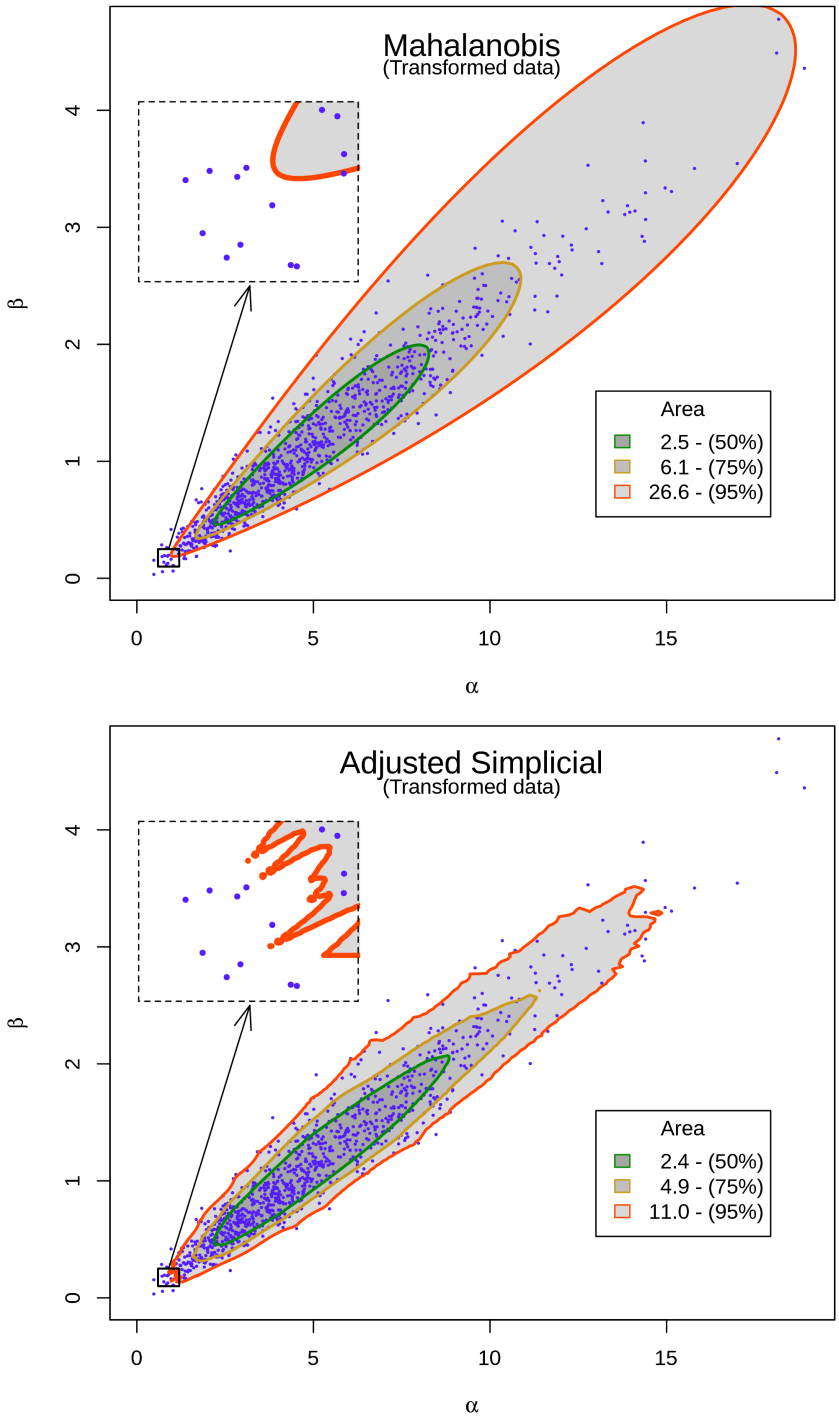

Figure 5.5: Plot of confidence regions constructed using the Mahalanobis depth (top) and the adjusted simplicial depth (bottom), after using the logarithmic transformation on the data sample and transforming the calculated region back to the original space. Three different confidence levels are used: 95% (red border), 75% (yellow border) and 50% (green border). The corresponding areas of the confidence regions are listed in the legend. The rectangle in the upper left corner is a zoomed in picture of the small rectangle in the bottom left corner.

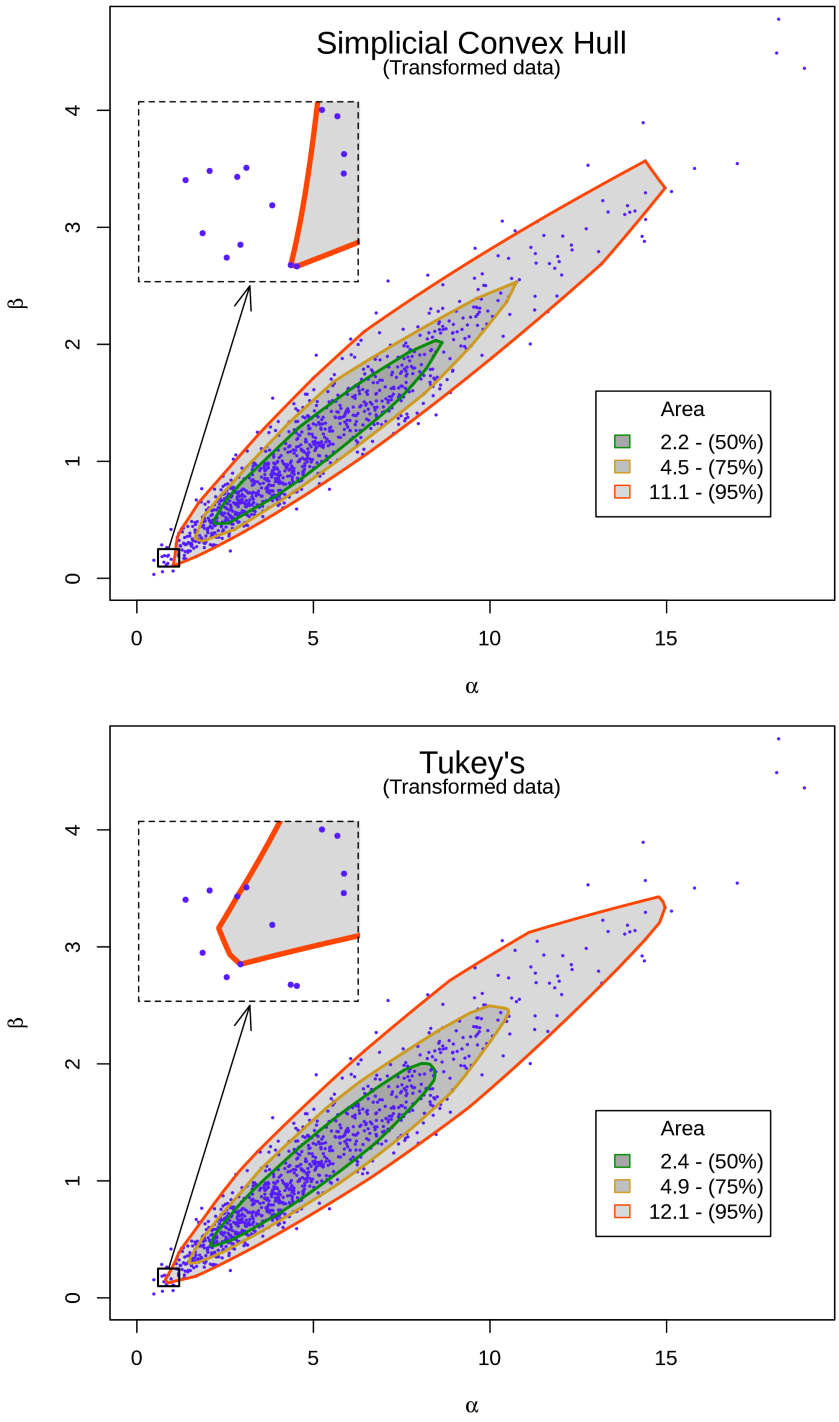

Figure 5.6: Plot of confidence regions constructed using the simplicial depth with convex hull (top) and the Tukey's depth (bottom), after using the logarithmic transformation on the data sample and transforming the calculated region back to the original space. Three different confidence levels are used: 95% (red border), 75% (yellow border) and 50% (green border). The corresponding areas of the confidence regions are listed in the legend. The rectangle in the upper left corner is a zoomed in picture of the small rectangle in the bottom left corner.

points in $(\tilde{\alpha}, \tilde{\beta})$. The results from this is shown in Figure 5.7. Here the logarithm used is in base 2, as before, and the largest number of iterations used in the plots is $2^{11} = 2\,048$. The desired confidence level is set to 95%.

The same is done using logarithmic transformation on $(\tilde{\boldsymbol{\alpha}}, \tilde{\boldsymbol{\beta}})$ and checking if the true parameter values are inside the confidence regions constructed by this alternative approach. The results of this are shown in Figure 5.8.

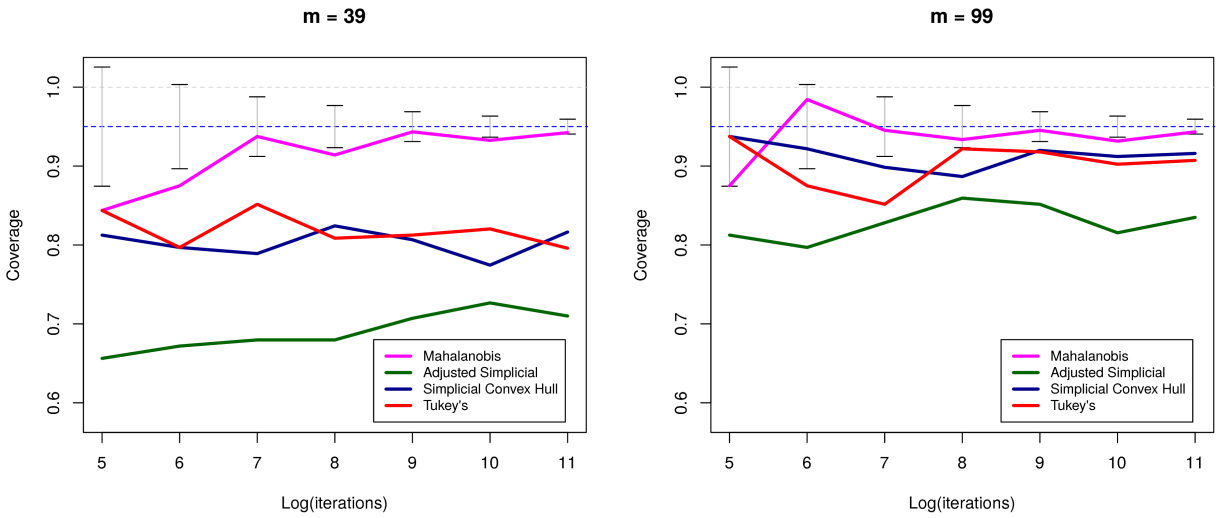

Figure 5.7: Coverage proportions of confidence regions for the parameters in the gamma distribution, using Algorithm 5 with a desired confidence level of 95%. The logarithm used on the x-axis has base 2.

5.4 Conclusion

Firstly, we observe from Figure 5.7 and 5.8 that the logarithmic transformation doesn't really affect how the coverage proportions. The shapes of the confidence regions, however, change quite a lot in some cases, as can be seen from Figure 5.2 and 5.5. The Mahalanobis depth gives a region that suits the distribution of the parameters better, since there is no extension to the negative numbers, although the area increases a lot. The regions constructed using the adjusted simplicial depth obtains a more smooth boundary when using the logarithmic transformation, but with almost no change in the area. We conclude by saying that the logarithmic transform

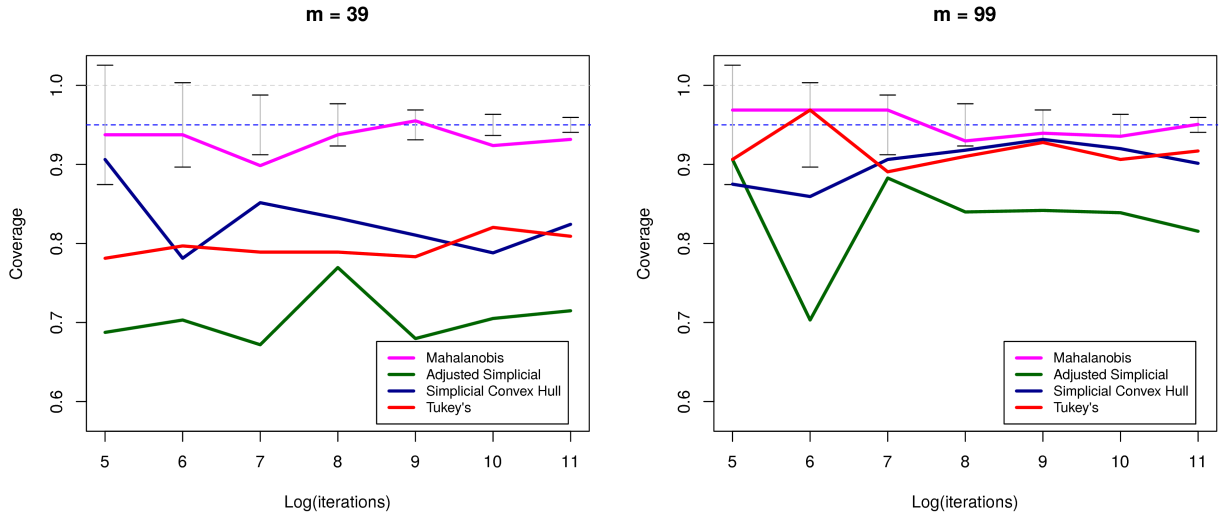

Figure 5.8: Coverage proportions of confidence regions for the parameters in the gamma distribution, using Algorithm 5 with a desired confidence level of 95%. Here the logarithmic transform is used on $(\tilde{\boldsymbol{\alpha}}, \tilde{\boldsymbol{\beta}})$. The logarithm used on the x-axis has base 2.

doesn't really help us much, and that it probably just adds unnecessary complexity to the algorithm.

The Mahalanobis depth actually gives close to correct coverage proportions for such small values as $n = 39$ and $n = 99$. This is likely because the distribution of $(\tilde{\boldsymbol{\alpha}}, \tilde{\boldsymbol{\beta}})$, shown in Figure 5.1, doesn't differ too much from that of a bivariate normal distribution. The draw back is the shape and area of the regions, as discussed earlier. It is also possible that the distribution of $(\tilde{\boldsymbol{\alpha}}, \tilde{\boldsymbol{\beta}})$ will have a different kind of shape for data samples $\mathbf{x}$, and that the ellipses generated using the Mahalanobis depth will be a poor fit for the actual underlying distribution.

The confidence region that combines a good coverage proportion with a simple shape, is the one generated using the Tukey's depth. It is also relatively fast to compute, using the depth-library in R described earlier. In total, this makes it a very attractive method for generating confidence regions for the parameters in the gamma distribution.

Bibliography

Bærentzen, J. A., Gravesen, J., Anton, F., and Aanæs, H. (2012). *Guide to Computational Geometry Processing: Foundations, Algorithms, and Methods*. Springer London.

Casella, G. and Berger, R. L. (2002). *Statistical Inference*. Brooks/Cole, Cengage Learning, 2nd edition.

Efron, B. (1979). Bootstrap methods: Another look at the jackknife. *The Annals of Statistics*, 7(1):1–26.

Hall, P. (1992). *The Bootstrap and Edgeworth Expansion*. Springer-Verlag New York.

Husak, G. J., Michaelsen, J., and Funk, C. (2007). Use of the gamma distribution to represent monthly rainfall in africa for drought monitoring applications. *International Journal of Climatology*, 27(7):935–944.

Iliopoulos, G. (2016). Exact confidence intervals for the shape parameter of the gamma distribution. *Journal of Statistical Computation and Simulation*, 86(8):1635–1642.

Lillegard, M. and Engen, S. (1999). Exact confidence intervals generated by conditional parametric bootstrapping. *Journal of Applied Statistics*, 26(4):447–459.

Lindqvist, B. H. and Taraldsen, G. Unpublished notes on the gamma distribution.

Liu, R. Y. (1990). On a notion of data depth based on random simplices. *The Annals of Statistics*, 18(1):405–414.

Liu, R. Y. and Singh, K. (1997). Notions of limiting p values based on data depth and bootstrap. *Journal of the American Statistical Association*, 92(437):266–277.

Mahalanobis, P. C. (1936). On the generalized distance in statistics. *Proceedings of the National Institute of Sciences (Calcutta)*, 2:49–55.

Ross, S. (2009). *Probability and Statistics for Engineers and Scientists*. Elsevier Academic Press, 4th edition.

Rousseuw, P. and Ruts, I. (1996). Algorithm as 307: Bivariate location depth. *Journal of the Royal Statistical Society Series C Applied Statistics*, 45:516–526.

Tukey, J. W. (1975). Mathematics and the picturing of data. In *Proceedings of the international congress of mathematicians*, volume 2, pages 523–531.

Yeh, A. B. and Singh, K. (1997). Balanced confidence regions based on tukey's depth and the bootstrap. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 59(3):639–652.

Zuo, Y. and Serfling, R. (2000). General notions of statistical depth function. *Annals of statistics*, pages 461–482.