ΟΙΚΟΝΟΜΙΚΟ ΠΑΝΕΠΙΣΤΗΜΙΟ ΑΘΗΝΩΝ
ATHENS UNIVERSITY OF ECONOMICS AND BUSINESS

SCHOOL OF INFORMATION SCIENCES & TECHNOLOGY

DEPARTMENT OF STATISTICS

POSTGRADUATE PROGRAM

A comparison of Monte Carlo goodness of fit procedures

By

Spyridon Sofianos

A THESIS

Submitted to the Department of Statistics

of the Athens University of Economics and Business

in partial fulfilment of the requirements for

the degree of Master of Science in Statistics

Athens, Greece

June 2016

ΟΙΚΟΝΟΜΙΚΟ ΠΑΝΕΠΙΣΤΗΜΙΟ ΑΘΗΝΩΝ
ATHENS UNIVERSITY OF ECONOMICS AND BUSINESS

ΣΧΟΛΗ ΕΠΙΣΤΗΜΩΝ & ΤΕΧΝΟΛΟΓΙΑΣ ΤΗΣ ΠΛΗΡΟΦΟΡΙΑΣ

ΤΜΗΜΑ ΣΤΑΤΙΣΤΙΚΗΣ

ΜΕΤΑΠΤΥΧΙΑΚΟ

Σύγκριση Monte Carlo διαδικασιών καλής προσαρμογής

Σπυρίδων Δημητρίου Σοφιανός

ΔΙΑΤΡΙΒΗ

Που υποβλήθηκε στο Τμήμα Στατιστικής

του Οικονομικού Πανεπιστημίου Αθηνών

ως μέρος των απαιτήσεων για την απόκτηση

Μεταπτυχιακού Διπλώματος Ειδίκευσης στη Στατιστική

Αθήνα

Ιούνιος 2016

DEDICATION

Dedicate to my family and friends.

ACKNOWLEDGEMENTS

I would like to thank Mr Panayiotis Besbeas for his guidance and help and my family for their support.

II

VITA

Born in Athens, Greece. I took my bachelor degree in Mathematics from the National and Kapodistrian University of Athens and my first Masters degree in actuarial mathematics from Economic University of Athens.

ABSTRACT

Spyridon Sofianos

A comparison of Monte Carlo goodness of fit procedures

June 2016

Goodness of fit is an important part of inference. Standard approaches such as chi-square method and bootstrap are asymptotic or highly time consuming. In this thesis we evaluate a new method of calibrated simulation proposed by Besbeas and Morgan (2014). We explore a new variant of the method and we compare the method against the bootstrap. The approaches of chi-square, bootstrap and calibrated simulation to check the goodness of fit of models are introduced and illustrated using real data.

ΠΕΡΙΛΗΨΗ

Σπυρίδων Σοφιανός

Σύγκριση Monte Carlo διαδικασιών καλής προσαρμογής

Ιούνιος 2016

Ο έλεγχος καλής προσαρμογής είναι ένα σημαντικό κομμάτι της στατιστικής επιστήμης. Κλασσικές προσεγγίσεις είναι το κριτήριο χ^2 και η μέθοδος bootstrap. Και οι δύο μέθοδοι είναι ασυμπτωτικές και χρειάζονται πολύ χρόνο για να παράξουν αποτελέσματα. Σε αυτή την εργασία εφαρμόζουμε μια καινούργια μέθοδο ρυθμιζόμενη προσομοίωσης που προτάθηκε από τους Besbeas and Morgan (2014). Θα εξερευνήσουμε μια καινούργια παραλλαγή της μεθόδου και θα την συγκρίνουμε με αυτή του bootstrap. Οι μέθοδοι του κριτήριου χ^2, του bootstrap, και της ρυθμιζόμενης προσομοίωσης για να ελέγξουμε την καλή προσαρμογή μοντέλων παρουσιάζονται και αναπτύσονται παρακάτω χρησιμοποιώντας πραγματικά δεδομένα.

TABLE OF CONTENTS

Page

(Continued)

LIST OF TABLES

LIST OF FIGURES

CHAPTER 1
INTRODUCTION

The main purpose of this thesis is to examine different methods for judging how well a model fits the data. This is known as goodness of fit and it is an important part of inference. Standard approaches for goodness of fit are the chi-square method and the use of the bootstrap. The chi-square method is asymptotic and suffers from the need to merge cells in order to have larger frequencies but by doing so we lose information while for the bootstrap the models have to be fitted to each of many different simulated data sets, which is highly time consuming.

Besbeas and Morgan (2014) propose and evaluate a new method of calibrated simulation. Here comparative data sets are obtained from simulating data when model parameter values are obtained from assumed asymptotic normal distribution of the maximum likelihood estimators from the real data. The approach is motivated and justified by Bayesian p-values. It limits the additional model-fitting that is required, and an improvement in efficiency is obtained relative to the bootstrap. Calibration of the resulting statistics is achieved as repeated data sets are easily simulated from the fitted model. The method requires the specification of model discrepancy measures. The approaches of chi-square, bootstrap and calibrated simulation to check the goodness of fit of models are introduced and illustrated using a variety of real data sets which arise in fecudability studies and capture-recapture.

In chapter 2 we have a thorough presentation of a fertility problem with real life data. The data here describe the number of fertility cycles to conception required by fertile human couples setting out to conceive. Since the couples in this study are essentially waiting for an event, the simplest probability model for waiting times when they are integer, is the geometric model. In practice the probability of conception of each

woman might be different, which gives rise to an hierarchical model the beta-geometric model. These two models are thoroughly presented.

In chapter 3 we do the model fitting and we are checking the chi-square goodness-of-fit. In order to do that we present analytically the maximum likelihood estimator hessian matrix, variance, standard error, the log-likelihood, the multinomial log-likelihood and finally the chi-square statistic for both geometric and beta-geometric models and we check the goodness-of-fit.

In chapter 4 we are going to see the historical background of bootstrap methods. A definition is also given and is calculated for both the geometric and beta-geometric models. We next use the parametric bootstrap to generate 500 samples and we calculate the deviances for each sample. We make a histogram to see where the observed deviance is located relative to the deviances from the simulated samples. We do this procedure for both models.

In chapter 5 we illustrate the use of calibrate simulation to examine how well the geometric and the beta-geometric distributions fit the women non-smokers data set In detail, suppose that $\hat{\theta}$ and $\hat{\Sigma}$ are respectively the maximum-likelihood estimates from fitting the real data, and associated dispersion matrix obtained from inverting the observed information matrix evaluated at $\hat{\theta}$. For each simulated parameter value $\theta_i \sim$ N $(\hat{\theta}, \hat{\Sigma})$ we might calculate a measure of the discrepancy between the data, $\boldsymbol{x}$ and the corresponding model, D($\boldsymbol{x}$;θ_i), and for each simulated parameter value θ_i we also simulate a new data set $\boldsymbol{x_i}$. For each new data set we then calculate D($\boldsymbol{x_i}$; θ_i), and a scatter plot is obtained of D($\boldsymbol{x_i}$; θ_i) vs D($\boldsymbol{x}$;θ_i). If the model fits the data well then one would expect approximately half of the points in the scatter plot to be above the line of unit slope through the origin.

In chapter 6 we examine the relative performance of the bootstrap and simulated calibration methods for evaluating the goodness of fit of a model, and we present the results using scatter-plots.

In chapter 7 we make an introduction to capture-recapture models. We examine the goodness-of-fit of model M_0, which is the simplest model where capture probability is constant over the capture occasions, and model M_t which the capture probabilities to vary with time. We used real life data to examine the goodness-of-fit for both models.

CHAPTER 2

FERTILITY PROBLEM

Reported falls in human sperm counts in many developed countries have serious implications for the future of mankind. In fecundability studies, data are collected on waiting times to conception in human beings, as well as on variables such as age and body mass index, which is a measure of obesity. For instance, the paper by Jensen et al. (1998) concluded that the probability of conception in a menstrual cycle was lowered if only five alcoholic drinks were taken by the woman each week. Data from studies such as this require appropriate statistical analysis, which quite often results from describing the data by means of models tailored specifically to the particular question of interest.

Table 2.1 describes the numbers of fertility cycles to conception required by fertile human couples setting out to conceive. The data were collected retrospectively, which means that information was only obtained from women who had conceived, and the women involved have been classified according to whether they smoked or not. Couples requiring more than 12 cycles are grouped together in a single category.

The couples in this study are essentially waiting for an event and the simplest probability model for waiting times when they are integer, as here, is the geometric model, we denote the geometric model as Model 1. Let X denote the number of cycles and let p be the probability of conception per cycle then

$$Pr(X = k) = (1 - p)^{k-1}p, \text{ for } k \geq 1. \quad (1)$$

In practice the probability of conception of each woman might be different, which gives rise to alternative models. The beta-geometric distribution arises as a hierarchical model (or infinite mixture) under the assumption of individual heterogeneity. We denote the beta-geometric model as Model 2. Let X denote the number of cycles and let $p \sim \text{Be}(\alpha, \beta)$ be the probability of conception per cycle then

$$Pr(X = k) = \frac{B(\alpha+1,\beta+k-1)}{B(\alpha,\beta)}, \text{ for } k \geq 1,$$

where B(α, β) is the beta function.

Table 2.1 Cycles to conception, classified by whether the female of the couple smoked or not. The data, taken from Weinberg and Gladen (1986), form a subset of data presented by Baird and Wilcox (1985). Excluded were women whose most recent method of contraception was the pill, as prior pill usage is believed to reduce fecundability temporarily. The definition of "smoking" is given in the source papers.

Cycle	Women non-smokers	Women smokers
1	198	29
2	107	16
3	55	17
4	38	4
5	18	3
6	22	9
7	7	4
8	9	5
9	5	1
10	3	1
11	6	1
12	6	3
>12	12	7
Total	486	100

We could assume that there is a third model which could be a mixture of two, or more generally k, geometric distributions. The assumption here would be that we have k groups of women with respect to probability p.

We are going to use the "non-smokers" data in order to examine how well the geometric and the beta-geometric models fit the data. The same procedure has been done with the "smokers" data also but the sample size is much smaller for these data, and the results are omitted.

CHAPTER 3

Model fitting and chi-square goodness-of-fit

3.1 Maximum likelihood estimation

Let $\mathbf{y} = (y_1, y_2, \ldots, y_n)'$ be a vector of independent and identically distributed (iid), random variables from one of a family of distributions on $\Re^n$ and indexed by a p-dimensional parameter $\boldsymbol{\theta} = (\theta_1, \ldots, \theta_p)'$ where $\boldsymbol{\theta} \in \Omega \subset \Re^p$ and $p \leq n$. Denote the distribution function of y by $F(y|\boldsymbol{\theta})$ and assume that the density function $f(y|\boldsymbol{\theta})$ exists. Then the likelihood function of $\boldsymbol{\theta}$ is given by:

$$L(\boldsymbol{\theta}) = \prod_{i=1}^{n} f(y_i \mid \boldsymbol{\theta}).$$

In practice, the natural logarithm of the likelihood function, called the log-likelihood function is denoted by:

$$\ell(\boldsymbol{\theta}) = \log L(\boldsymbol{\theta}) = \sum_{i=1}^{n} \log f(y_i \mid \boldsymbol{\theta}),$$

is used since it is found to be easier to manipulate algebraically. Let the p partial derivatives of the log-likelihood form the $p \times 1$ vector

$$\mathbf{u}(\boldsymbol{\theta}) = \frac{\partial \ell(\boldsymbol{\theta})}{\partial \boldsymbol{\theta}} = \begin{pmatrix} \frac{\partial \ell}{\partial \theta_1} \\ \vdots \\ \frac{\partial \ell}{\partial \theta_p} \end{pmatrix}$$

The vector $\mathbf{u}(\boldsymbol{\theta})$ is called the score vector of the log-likelihood function. The moments $\mathbf{u}(\boldsymbol{\theta})$ satisfy two important identities. First, the expectation of $\mathbf{u}(\boldsymbol{\theta})$ with respect to y is equal to zero, and second, the variance of $\mathbf{u}(\boldsymbol{\theta})$ is the negative of the expectation of the second derivative of $\ell(\boldsymbol{\theta})$, i.e.,

$$\text{Var}\,(\mathbf{u}(\boldsymbol{\theta})) = -\text{E}\{\mathbf{u}(\boldsymbol{\theta})\boldsymbol{u}(\boldsymbol{\theta})^{\boldsymbol{T}}\} = \left(-\text{E}\left\{\left(\frac{\partial^2 \ell(\boldsymbol{\theta})}{\partial \theta_j \partial \theta_\kappa}\right)\right\}\right).$$

The $p \times p$ matrix on the right hand side is called the expected Fisher information matrix and usually denoted by $\mathfrak{T}(\boldsymbol{\theta}) = \left(- \mathrm{E}\left\{ \left(\frac{\partial^2 \ell(\boldsymbol{\theta})}{\partial \theta_j \partial \theta_\kappa} \right) \right\} \right)$. The expectation here is taken over the distribution of y at a fixed value of $\boldsymbol{\theta}$. The maximum likelihood estimate of $\boldsymbol{\theta}$ is given by the solution $\widehat{\boldsymbol{\theta}}$ to the p equations

$$\mathbf{u}(\widehat{\boldsymbol{\theta}}) = 0$$

and under some regularity conditions, the distribution of $\widehat{\boldsymbol{\theta}}$ is asymptotically normal with mean $\boldsymbol{\theta}$ and variance covariance matrix given by the $p \times p$ matrix $\mathfrak{T}(\boldsymbol{\theta})^{-1}$i.e., the inverse of the expected information matrix. The $p \times p$ matrix

$$\mathbf{I}(\boldsymbol{\theta}) = -\left\{ \frac{\partial^2 \ell(\boldsymbol{\theta})}{\partial \theta_j \partial \theta_\kappa} \right\}$$

is called the observed information matrix. In practice, since the true value of $\boldsymbol{\theta}$ is not known, these two matrices are estimated by substituting the estimated value $\widehat{\boldsymbol{\theta}}$ to give $\mathfrak{T}(\widehat{\boldsymbol{\theta}})$ and $\mathbf{I}(\widehat{\boldsymbol{\theta}})$, respectively. Asymptotically, these forms of the information matrix can be shown to be equivalent because the $\mathfrak{T}(\widehat{\boldsymbol{\theta}})$ and $\mathbf{I}(\widehat{\boldsymbol{\theta}})$ are the maximum likelihood estimators of $\mathfrak{T}(\boldsymbol{\theta})$ and $\mathbf{I}(\boldsymbol{\theta})$ respectively.

From a computational standpoint, the above quantities are related to those computed to solve an optimization problem as follows: $-\ell(\boldsymbol{\theta})$ corresponds to the objective function to be minimized, $\mathbf{u}(\boldsymbol{\theta})$ represents the gradient vector, the vector of first order partial derivatives and $\mathbf{I}(\boldsymbol{\theta})$, corresponds to the negative of the Hessian matrix $H(\boldsymbol{\theta})$, the matrix of second-order derivatives of the objective function respectively:

$$-H(\boldsymbol{\theta}) = -\frac{\partial^2 \ell(\boldsymbol{\theta})}{\partial \theta_j \partial \theta_\kappa}$$

In the MLE problem, the Hessian matrix is used to determine whether the solution $\widehat{\boldsymbol{\theta}}$ to the equations $\mathbf{u}(\boldsymbol{\theta})=\mathbf{0}$ corresponds to a minimum of the objective function $-\ell(\boldsymbol{\theta})$ but more importantly it is used through the observed information matrix $\mathbf{I}(\widehat{\boldsymbol{\theta}})$ for estimating the asymptotic covariance matrix of $\widehat{\boldsymbol{\theta}}$, since if $\mathfrak{T}(\widehat{\boldsymbol{\theta}})$ were to be used then the expectation of $\mathbf{I}(\widehat{\boldsymbol{\theta}})$ needs to be evaluated analytically. Thus $\mathrm{Var}(\widehat{\boldsymbol{\theta}}) = \left[\mathbf{I}\left(\widehat{\boldsymbol{\theta}}\right)\right]^{-1} = [-H\left(\widehat{\boldsymbol{\theta}}\right)]^{-1}$. Moreover, if computing the derivatives of $\ell(\boldsymbol{\theta})$ in closed form is difficult or if the optimization procedure does not produce an estimate of the Hessian

as a byproduct, estimates of the derivatives obtained using finite difference methods may be substituted for $\mathbf{I}(\widehat{\boldsymbol{\theta}})$.

As we shall see in the next sections, the standard errors of the estimators $\widehat{\boldsymbol{\theta}}$, are just the square roots of the diagonal terms of the variance–covariance matrix Var($\widehat{\boldsymbol{\theta}}$). The maximum likelihood estimates, hessian matrices and standard errors have been computed with the help of the R-programming language.

3.2 Chi-square test

The most famous test for checking the validity of a distribution to describe a random phenomenon based on a set of experimental data is the chi-square test. The test evaluates the null hypothesis H_0 that the data are governed by the assumed distribution, against the alternative H_1 that the data are not drawn from the assumed distribution:

H_0 : the data are governed by the assumed distribution.

H_1: the data are not drawn from the assumed distribution.

Let $p_1, p_2, \dots, p_k$ denote the probabilities hypothesized for k possible categories under H_0. In n independent trials, we let $n_1, n_2, \dots, n_k$ denote the observed frequencies (O_i, $1 \leq i \leq k$) of each outcome which are to be compared to the expected frequencies np_1, $np_2, \dots, np_k$, (E_i, $1 \leq i \leq k$).

The test is based on the chi-square test statistic which is defined as: $X^2 = \sum_{i=1}^{k} \frac{(O_i - E_i)^2}{E_i}$.

The standardized counts $\sqrt{\frac{(O_i - E_i)^2}{E_i}}$ for k categories are approximately normal, but they are not independent because one of the counts is entirely determined by the sum of the others (since the total of the observed and the expected counts must sum to n). This results in a loss of one degree of freedom, so it turns out the distribution of the chi-squared test statistic based on k counts is approximately the chi-squared

distribution with $m = k - 1$ degrees of freedom, denoted X^2_{k-1}. But often the assumed distribution has unknown parameters which we have to estimate. In this case the asymptotic distribution of the chi-square test statistic is the X^2 with (k − 1 − r) degrees of freedom, where k are the number of counts and r the number of parameters which we have estimated. There are some restrictions which we have to take into account:

a) All the expected frequencies $E_i \geq 1$
b) And a maximum 20% of the expected frequencies are $E_i \leq 5$

or

a) All the expected frequencies are $E_i \geq 5$

In order to continue we shall take into account that all the expected frequencies are $E_i \geq 5$.

3.3 Goodness of fit testing that the data follow a geometric distribution.

3.3.2 Computation of simple log-likelihood, hessian matrix, variance and standard error

Let X denotes the number of cycles to conception. In this section we assume that X has a geometric distribution with probability function given by Equation (1).

The likelihood function for a set of observations **x** = (x_1,…,x_n) is given by

$$L(\boldsymbol{x} \mid p) = \Pi_{i=1}^{n} p(1-p)^{(x_i-1)} = p^n(1-p)^{(\Sigma_{i=1}^{n} x_i - n)}$$

and the log-likelihood is given by

$$\ell\,(p) = nlog(p) + (\sum_{i=1}^{n} x_i - n)log(1-p).$$

And it is straightforward to show that the maximum likelihood estimator of p is given by: $\hat{p} = \frac{n}{\sum_{i=1}^{n} x_i}$,

For the non-smokers data set, the log-likelihood (which we are going to call simple log-likelihood in order to distinguish it from the multinomial log-likelihood below) is given by

$$\ell\,(p) \;=\; 486log(p) \;+\; (1441 - 486)log(1-p)$$

resulting in the maximum likelihood estimate $\hat{p} = \frac{n}{\sum_{i=1}^{n} x_i} = \frac{486}{1441} = 0.337$ or 33.7%.

This assumes that $\sum_{i=1}^{n} x_i$ = 198·1 + 107·2 + 55·3 + 38·4 + 18·5 + 22·6 + 7·7 + 9·8 + 5·9 + 3·10 + 6·11 + 6·12 + 12·13= 1441 and which is obtained by setting all the observed cycles above 12 equal to 13 this is why the standard (simple) log-likelihood is not very appropriate.

The derivatives of the log-likelihood with respect to p are:

$$\frac{\partial \ell\,(\mathrm{p})}{\partial p} = \frac{n}{p} - \frac{(\sum_{i=1}^{n} x_i - n)}{1-p}$$

$$\frac{\partial^2 \ell\,(\mathrm{p})}{\partial p^2} = -\frac{n}{p^2} - \frac{(\sum_{i=0}^{n} x_i - n)}{(1-p)^2}$$

And thus the observed information matrix, $\mathbf{I}(p) = -\left(\frac{\partial^2 \ell\,(\mathrm{p})}{\partial p^2}\right)$, evaluated at $p = \hat{p}$ is equal to:

$$\mathbf{I}(\hat{p}) = -\left(\frac{-486}{0.337^2} - \frac{(1441-486)}{(1-0.337)^2}\right) = 6451.919.$$

The expected information matrix is given by:

$$\mathfrak{T}\,(p) = -\mathrm{E}\left\{\frac{\partial^2 \ell\,(\mathrm{p}))}{\partial p^2}\right\} = \mathrm{E}\left\{\frac{n}{p^2} + \frac{(\sum_{i=1}^{n} x_i - n)}{(1-p)^2}\right\} = \frac{n}{p^2} + \frac{n(\frac{1}{p}-1)}{(1-p)^2} = \frac{n}{p^2(1-p)}.$$

and equals $\mathfrak{T}\,(\hat{p})$ = 6451.919 for the non-smokers data.

The hessian matrix is $\mathrm{H}(\hat{p}) = \frac{\partial^2 \ell\,(\mathrm{p})}{\partial p^2}$ = -6451.919 and thus the variance and standard error are:

$$\mathrm{Var}(\hat{p}) = [\mathbf{I}(\hat{\mathrm{p}})]^{-1} = (-[H(\hat{p})])^{-1} = \left(-\left[\frac{\partial^2 \ell\,(\mathrm{p})}{\partial p^2}\right]\right)^{-1} = (6451.919)^{-1}$$

Standard error = $\sqrt{(6446.462)^{-1}}$= 0.01245487.

3.3.3 Computation of multinomial log-likelihood, hessian matrix, variance and standard error

The simple likelihood assumes the observed cycles above 12 are all equal to 13. We can avoid arbitrarily setting the unknown number of cycles above 12 to 13 by assuming a multinomial likelihood structure.

Multinomial likelihood function:

L(**x**|p) = $\frac{n!}{n_1!n_2!.....n_u!}\prod_{i=1}^{u} p_i{}^{n_i}$, 1≤i≤u, (u=13, for this example)

Where n_i denote the observed frequencies and $p_i = P(X = i) = (1-p)^{i-1}p$, for $1 \leq i < 13$ and

$p_{13} = P(x \geq 13) = 1 - (\sum_{i=1}^{12} P(X = i))$=1-(1-$(1-p)^{12}$)=$(1-p)^{12}$.

The log-likelihood is given by ℓ (p) = c + ($\sum_{i=1}^{u} n_i \log (p(1-p)^{i-1})$)

where c is a constant that does not depend upon p. The derivatives with respect to p are :

$$\frac{\partial \ell\,(\mathrm{p})}{\partial p} = \frac{\sum_{i=1}^{u-1} n_i}{p} - \frac{\sum_{i=1}^{u-1} n_i(i-1)}{1-p} - \frac{n_u(u-1)}{1-p}$$

$$\frac{\partial^2 \ell\,(\mathrm{p})}{\partial p^2} = -\frac{\sum_{i=1}^{u-1} n_i}{p^2} - \frac{\sum_{i=1}^{u-1} n_i(i-1)}{(1-p)^2} - \frac{n_u(u-1)}{(1-p)^2}.$$

Setting the score to zero results in the maximum likelihood estimate

$$\hat{p} = \frac{\sum_{i=1}^{u-1} n_i}{\sum_{i=1}^{u-1} n_i(i-1) + n_u(u-1) + \sum_{i=1}^{u-1} n_i} = \frac{474}{474+144+811} = 0.3317.$$

The observed information matrix is, $\mathbf{I}(\boldsymbol{p})$ = - $(\frac{\partial^2 \ell\,(\mathrm{p})}{\partial p^2})$, evaluated at $p = \hat{p}$ is

$$\mathbf{I}(\hat{p}) = -\left(-\frac{474}{0.11} - \frac{811}{0.447} - \frac{144}{0.447}\right) = 6446.462.$$

The hessian matrix is $H(\hat{p}) = \frac{\partial^2 \ell(p)}{\partial p^2} =$ -6446.462 and thus the variance and standard error are:

$$Var(\hat{p}) = [\mathbf{I}(\hat{\mathbf{p}})]^{-1} = (-[H(\hat{p})])^{-1} = \left(-\left[\frac{\partial^2 \ell(p))}{\partial p^2}\right]\right)^{-1} = (6446.462)^{-1}$$

Standard error = $\sqrt{(6446.462)^{-1}} = 0.01245487$.

We observe that there is little difference in the maximum likelihood estimate for p between simple and multinomial likelihoods.

3.4 Chi-squared test

We evaluate the goodness of fit of the geometric distribution using a chi-squared test. The chi-square statistic is given by:

$$X^2 = \sum_{i=1}^{10} \frac{(O_i - E_i)^2}{E_i} \sim X^2_{8,1-a}$$

Table 3.1 gives the expected values under the geometric distribution fitted by maximum likelihood.

Cycle	Observed values	Simple likelihood	Multinomial likelihood
1	198	163.9	161.2
2	107	108.6	107.7
3	55	72	72
4	38	47.7	48.1
5	18	31.6	32.2
6	22	21	21.5
7	7	13.9	14.4
8	9	9.2	9.6
9	5	6.1	6.4
10	3	4	4.3
11	6	2.7	2.9

12	6	1.8	1.9
>12	12	3.5	3.9
Total	Total	486	486

Table 3.1: expected values under a geometric distribution when p is estimated by simple and multinomial likelihood.

The expected values were calculated by np_j, $1 \leq j \leq 13$ where $n = 486$ and p_j denotes the probabilities of the 13 categories, and the p_j's are calculated in two ways, first using the M.L.E. from the simple log-likelihood and the second time the M.L.E. from the multinomial log-likelihood. Since there are expected frequencies below 5 we are going to amalgamate the groups so we are going to have k = 10 groups instead of 13. In the matrix above we can see that the last 4 groups have expected frequencies less than 5 so we are going to take them into consideration as 1 group, resulting in 10-1-1=8 degrees of freedom.

For the simple log-likelihood:

$$\sum_{i=1}^{10} \frac{(O_i-E_i)^2}{E_i} = \frac{(198-163.9)^2}{163.9} + \frac{(107-108.6)^2}{108.6} + \frac{(55-72)^2}{72} + \frac{(38-47.7)^2}{47.7} + \frac{(18-31.6)^2}{31.6} + \frac{(22-21)^2}{21} + \frac{(7-13.9)^2}{13.9} + \frac{(9-9.2)^2}{9.2} + \frac{(5-6.1)^2}{6.1} + \frac{(27-12)^2}{12} = 41.38$$

$X^2 = \sum_{i=1}^{10} \frac{(O_i-E_i)^2}{E_i} \sim X^2_{8,1-a} = 15.51$, for α = 0.05 we have that 41.38>15.51 where 15.51 is the critical value of $X^2_{8,1-a}$ with α = 0.05. We reject the null hypothesis that the data are geometrically distributed.

For the multinomial log-likelihood:

$$\sum_{i=1}^{10} \frac{(O_i-E_i)^2}{E_i} = \frac{(198-161.2)^2}{161.2} + \frac{(107-107.7)^2}{107.7} + \frac{(55-72)^2}{72} + \frac{(38-48.1)^2}{48.1} + \frac{(18-32.2)^2}{32.2} + \frac{(22-21.5)^2}{21.5} + \frac{(7-14.4)^2}{14.4} + \frac{(9-9.6)^2}{9.6} + \frac{(5-6.4)^2}{6.4} + \frac{(27-12.9)^2}{12.9} = 40.37$$

$X^2 = \sum_{i=1}^{10} \frac{(O_i-E_i)^2}{E_i} \sim X^2_{8,1-a}$, for α = 0.05 we have that $40.37 > 15.51$ where 15.51 is the critical value of $X^2_{8,1-a}$ with α = 0.05. The result is that the null

hypothesis that the data are geometrically distributed is rejected and so this model is not appropriate for continuing our research.

From the above, we reach the conclusion that whatever likelihood we use simple or multinomial, the data are not distributed geometrically. There is the possibility that an alternative model would be a mixture of 2 geometric distributions but let's examine the possibility that the data are distributed beta-geometrically.

3.5 Goodness of fit testing that the data follow a beta-geometric distribution

In this section we examine the possibility that there is a variation in the probability of conception between women, resulting for example from individual heterogeneity. This gives rise to a hierarchical model but in this case we need to find the marginal distribution. So we define the Hierarchical Model:

$X \mid p \sim$ Geometric (p)

$p \sim f(p)$ and we focus on

$p \sim$ Beta(α,β)

$f(p) = \frac{1}{B(\alpha,\beta)} p^{\alpha-1}(1-p)^{\beta-1}$, α > 0 and β > 0

The marginal distribution for X which is calculated in the following steps:

$$P(X = x) = \int_0^1 f(x,p)dp = \int_0^1 f(x \mid p) f(p)dp$$

$$= \int_0^1 p(1-p)^{x-1} \frac{1}{B(\alpha,\beta)} p^{\alpha-1}(1-p)^{\beta-1}dp$$

$$= \frac{\Gamma(\alpha+\beta)}{\Gamma(\alpha)\Gamma(\beta)} \int_0^1 p^{(\alpha+1)-1}(1-p)^{(\beta+x-1)-1}dp$$

$$= \frac{\Gamma(\alpha+\beta)}{\Gamma(\alpha)\Gamma(\beta)} \frac{\Gamma(\alpha+1)\Gamma(\beta+x-1)}{\Gamma(\alpha+\beta+x)} = \frac{B(\alpha+1,\beta+x-1)}{B(\alpha,\beta)} \quad (x = 1,2,3,\dots),$$

where $B(\alpha,\beta) = \frac{\Gamma(\alpha)\Gamma(\beta)}{\Gamma(\alpha+\beta)}$, is the beta function.

This is known as the beta-geometric distribution.

3.5.1 Computation of simple log-likelihood, hessian matrix, variance-covariance matrix and standard error

Suppose that data are available on n individuals as x_i, $i = 1,2,\dots,n$. The likelihood function for data based on beta-geometric distribution with parameters $\boldsymbol{\theta}$=(α,β) is given by:

$$L(\boldsymbol{\theta})=\prod_{i=1}^{n}\frac{B(\alpha+1,\beta+x_i-1)}{B(\alpha,\beta)}$$

and the corresponding log-likelihood, $\ell(\boldsymbol{\theta})$, is given as

$$\ell(\boldsymbol{\theta}) = \sum_{i=1}^{n}\left(\log B(\alpha+1,\beta+x_i-1)\right) - n\log(B(\alpha,\beta)).$$

The components of the score vector $\mathbf{u}(\boldsymbol{\theta}) = \left(\frac{\partial\ell(\theta)}{\partial\alpha},\frac{\partial\ell(\theta)}{\partial\beta}\right)^T$ is given by:

$$\frac{\partial\ell(\theta)}{\partial\alpha} = N\cdot\psi(\alpha+1) + N\cdot\psi(\alpha+\beta) - \sum_{i=1}^{n}\psi(x_i+\alpha+\beta+1) - N\psi(\alpha)$$

$$\frac{\partial\ell(\theta)}{\partial\beta} = \sum_{i=1}^{n}\psi(x_i+\beta) + N\cdot\psi(\alpha+\beta) - \sum_{i=1}^{n}\psi(x_i+\alpha+\beta+1) - N\psi(b)$$

where $\psi(x) = \frac{\Gamma'(x)}{\Gamma(x)}$

The maximum likelihood estimates α and β can be obtained either by directly maximizing the log likelihood function with respect to θ or by solving the two simultaneous equations obtained by equating $\mathbf{u}(\boldsymbol{\theta})$ =0. The results are $\hat{\alpha}$ = 4.276 and $\hat{\beta}$ = 6.539. The mean and the variance of the fitted Beta are $E(p) = \frac{\hat{\alpha}}{\hat{\alpha}+\hat{\beta}}$ =0.395, $Var(p)$ = $\frac{\hat{\alpha}\hat{\beta}}{(\hat{\alpha}+\hat{\beta})^2(\hat{\alpha}+\hat{\beta}+1)}$ = 0.0202. There is a small difference in these values compared with the values from the geometric.

The hessian matrix from the simple log-likelihood is the 2 x 2 matrix:

$$\mathbf{H}(\hat{\theta}) = \begin{bmatrix} 17.338121 & -9.234814 \\ -9.234814 & 5.138331 \end{bmatrix}$$

and thus the variance – covariance matrix of $\hat{\boldsymbol{\theta}}$ =($\hat{\alpha}$, $\hat{\beta}$) is :

$Var(\hat{\boldsymbol{\theta}}) = \left(-\begin{bmatrix} 17.338121 & -9.234814 \\ -9.234814 & 5.138331 \end{bmatrix}\right)^{-1} = \begin{bmatrix} 1.349633 & 2.425615 \\ 2.425615 & 4.554029 \end{bmatrix}$. This results in the standard errors 1.161737 and 2.134017 for $\hat{\alpha}$ and $\hat{\beta}$ respectively:

	$\hat{\alpha}$ (standard error)	$\hat{\beta}$ (standard error)
Simple log-likelihood	4.28 (1.1617)	6.54(2.1340)

3.5.2 Computation of multinomial log-likelihood, hessian matrix, variance-covariance matrix and standard error

The simple likelihood assumes that the observed cycles above 12 are all equal to 13. We can avoid arbitrarily setting the unknown number of cycles above 12 to 13 by assuming a multinomial likelihood structure.

Multinomial likelihood function:

The log-likelihood is given by $\ell(\mathrm{p}) = c + (\sum_{i=1}^{u} n_i log\ (p_i))$ where the c is a constant which does not affect the maximization, n_i denote the observed frequencies and where the cell probabilities are defined by the marginal distribution of X:

$p_i = \frac{B(\alpha+1,\beta+x_i-1)}{B(\alpha,\beta)}$, for $1 \leq i < 13$ and

$p_{13} = P(x \geq 13) = 1 - (\sum_{i=1}^{12} P(X = i)) = 1 - \sum_{i=1}^{12} \frac{B(\alpha+1,\beta+x_i-1)}{B(\alpha,\beta)}$.

The optimization of the multinomial log-likelihood, through the BFGS method (Broyden, Fletcher, Goldfarb and Shanno method), gives the maximum likelihood estimates $\hat{\alpha} = 2.987$ and $\hat{\beta} = 4.333$ for the group of non smokers. The mean and the variance of the fitted Beta are $E(\hat{p}) = \frac{\hat{\alpha}}{\hat{\alpha}+\hat{\beta}} = 0.408$, $Var(\hat{p}) = \frac{\hat{\alpha}\hat{\beta}}{(\hat{\alpha}+\hat{\beta})^2(\hat{\alpha}+\hat{\beta}+1)} = 0.029$.
The hessian matrix from the multinomial log-likelihood is the 2 x 2 matrix:

$$\mathbf{H}(\hat{\theta}) = \begin{bmatrix} 34.6496 & -18.51060 \\ -18.51060 & 10.66369 \end{bmatrix},$$

and thus the variance – covariance matrix of $\hat{\boldsymbol{\theta}} = (\hat{\alpha}, \hat{\beta})$ now is :

$Var(\hat{\theta}) = \left(-\begin{bmatrix} 34.6496 & -18.51060 \\ -18.51060 & 10.66369 \end{bmatrix}\right)^{-1} = \begin{bmatrix} 0.3971559 & 0.6894045 \\ 0.6894045 & 1.2904816 \end{bmatrix}$. This results in the standard errors 0.6302 and 1.1352 for $\hat{\alpha}$ and $\hat{\beta}$ respectively:

	$\hat{\alpha}$ (standard error)	$\hat{\beta}$ (standard error)
Multinomial log-likelihood	2.987 (0.6302)	4.333 (1.1359)

The difference in the maximum likelihood estimates between simple and multinomial likelihoods is interesting.

3.6 Chi-squared test

As in the previous section, we evaluate the goodness of fit of the beta-geometric distribution using a chi-square test. Table 2 provides the expected values under the beta-geometric fitted by simple and multinomial likelihood. For both likelihoods, there are three expected frequencies below 5, which we amalgamate into one, so we shall have 11 groups minus 1 and minus 2 for the two parameters (α,β) resulting into 8 degrees of freedom.

Number of cycles	Observed values	Simple log-likelihood	Multinomial log-likelihood
1	198	192.2	198.4
2	107	106.4	103.3
3	55	62.6	59.1
4	38	38.7	36.3
5	18	24.9	23.5
6	22	16.6	15.9
7	7	11.4	11.1
8	9	8	8
9	5	5.8	5.9
10	3	4.2	4.5
11	6	3.2	3.5

12	6	2.4	2.7
>12	12	9.8	13.9
Total	486	486	486

Table 3.2: Women non-smokers observed and expected values under a beta-geometric distribution when parameters are estimated by simple and multinomial likelihood.

Thus for the simple log-likelihood:

$$\sum_{i=1}^{11}\frac{(O_i-E_i)^2}{E_i} = \frac{(198-192.2)^2}{192.2} + \frac{(107-106.4)^2}{106.4} + \frac{(55-62.6)^2}{62.6} + \frac{(38-38.7)^2}{38.7} + \frac{(18-24.9)^2}{24.9} + \frac{(22-16.6)^2}{16.6} + \frac{(7-11.4)^2}{11.4} + \frac{(9-8)^2}{8} + \frac{(5-5.8)^2}{5.8} + \frac{(15-9.8)^2}{9.8} + \frac{(12-9.8)^2}{9.8} = 9.7$$

$X^2 = \sum_{i=1}^{11}\frac{(O_i-E_i)^2}{E_i} \sim X^2_{8,1-a}$= 15.51 , for α = 0.05 since 9.7 < 15.51 where 15.51 is the critical value of $X^2_{8,1-a}$ with α = 0.05. The result is that the null hypothesis, that the data are beta-geometrically distributed, is not rejected.

For multinomial log-likelihood:

$$\sum_{i=1}^{11}\frac{(O_i-E_i)^2}{E_i} = \frac{(198-198.4)^2}{198.4} + \frac{(107-103.3)^2}{103.3} + \frac{(55-59.1)^2}{59.1} + \frac{(38-36.3)^2}{36.3} + \frac{(18-23.5)^2}{23.5} + \frac{(22-15.9)^2}{15.9} + \frac{(7-11.1)^2}{11.1} + \frac{(9-8)^2}{8} + \frac{(5-5.9)^2}{5.9} + \frac{(15-10.7)^2}{10.7} + \frac{(12-13.9)^2}{13.9} = 6.721$$

$X^2 = \sum_{i=1}^{11}\frac{(O_i-E_i)^2}{E_i} \sim X^2_{8,1-a}$ =15.51, for α = 0.05 since 6.721 < 15.51 where 15.51 is the critical value of $X^2_{5,1-a}$ with α = 0.05. The result is that we accept the null hypothesis that the beta-geometric fits the data.

CHAPTER 4

Using bootstrap methods to test the goodness of fit of a model

4.1 Historical background of bootstrap methods

The ‘’bootstrap’’ is one of a number of techniques that is now part of the broad umbrella of non-parametric statistics that are commonly called resampling methods Some of the techniques are far older than the bootstrap. Permutation methods go back to Fisher (1935) and Pitman (1937,1938), and the jackknife started with Quenouille (1949). Bootstrapping was made practical through the use of the Monte Carlo approximation, but it goes back to the beginning of computers in the early 1940s.

However, 1979 is a critical year for the bootstrap because that is when Brad Efron's paper in the Annals of Statistics was published (Efron, 1979). Efron had defined a resampling procedure that he coined as bootstrap. He constructed it as a simple approximation to the jackknife (an earlier resampling method that was developed by John Tukey), and his original motivation was to derive properties of the bootstrap to better understand the jackknife. However in many situations, the bootstrap is as good as or better than the jackknife as a resampling procedure. The jackknife is primarily useful for small samples, becoming computationally inefficient for larger samples but has become more feasible as computer speed increases. A clear description of the jackknife and its connection to the bootstrap can be found in the SIAM monograph Efron (1982).

Although permutation tests were known in the 1930s, an impediment to their use was the large number (i.e., $n!$) of distinct permutations available for samples of size n. Since ordinary bootstrapping involves sampling with replacement n times for a sample n, there are n^n possible distinct ordered bootstrap samples (though some are equivalent under the exchageability assumption because they are permutations of each other). So, complete enumeration of all the bootstrap samples becomes infeasible except in very small sample sizes. Random sampling from the set of possible bootstrap samples becomes viable way to approximate the distribution of bootstrap samples. The same problem exists for permutations and the same remedy is possible. The only difference is that $n!$ does not grow as fast as n^n, and complete enumeration of permutation is possible for larger n than for the bootstrap.

The idea of taking several Monte Carlo samples of size n with replacement from the original observations was certainly an important idea expressed by Efron but was clearly known and practiced prior to Efron (1979). Although it may not be the first time it was used, Julian Simon laid claim to priority for the bootstrap based on his use

of the Monte Carlo approximation in Simon (1969). But Simon was only recommending the Monte Carlo approach as a way to teach probability and statistics in a more understandable way. After Efron made the bootstrap popular, Simon and Bruce joined the campaign (see Simon and Bruce, 1991, 1995).

Efron, however, starting with Efron (1979), first connected bootstrapping to the jackknife, delta method, cross-validation, and permutation tests. He was the first to show it to be a real competitor to the jackknife and delta method for estimating the standard error of an estimator. Also quite early on, Efron recognized the broad applicability of bootstrapping for confidence intervals, hypothesis testing, and more complex problems. These ideas were emphasized in Efron and Gong (1983), Diaconis and Efron (1983), Efron and Tibshirani (1986), and the SIAM monograph (Efron 1982). These influential articles along with the SIAM monograph led to a great new deal of research during the 1980s and 1990s. The explosion of bootstrap papers grew at an exponential rate. Key probabilistic results appeared in Singh (1981), Bickel and Freedman (1981, 1984), Beran (1982), Martin(1990), Hall (1986, 1988), Hall and Martin (1988), and Navidi (1989).

In a very remarkable paper, Efron (1983) used simulation comparisons to show that the use of bootstrap bias correction could provide better estimates of classification error rate than the very popular cross-validation approach (often called leave-one-out and originally proposed by Lachenbruch and Mickey, 1968). These results applied when the sample size was small, and classification was restricted to two or three classes only, and the predicting features had multivariate Gaussian distributions. Efron compared several variants of the bootstrap with cross-validation and the resubstitution methods. This led to several follow up articles that widened the applicability and superiority of a version of the bootstrap called 632. See Chatterjee and Chattejee (1983), Chernick et al. (1985, 1986, 1988a,b), Jain et al. (1987), and Efron and Tibshirani (1997).

Chernick was a graduate student at Stanford in the late 1970s when the bootstrap activity began on the Stanford and Berkeley campuses. However, oddly the bootstrap did not catch on with many graduate students. Even Brad Efron's graduate students chose other topics for their dissertation. Gail Gong was the first student of Efron to do a dissertation on the bootstrap. She did very useful applied work on using the bootstrap in model building (particularly for logistic regression subset selection). See Gong (1986). After Gail Gong, a number of graduate students wrote dissertations on the bootstrap under Efron, including Terry Therneau, Rob Tibshirani, and Tim Hesterberg. Michael Martin visited Stanford while working on his dissertation on bootstrap confidence intervals under Peter Hall. At Berkeley, William Navidi did his thesis on bootstrapping in regression and econometric models under David Freedman.

While exciting theoretical results developed for the bootstrap in the 1980s and 1990s, there were also negative results where it was shown that the bootstrap estimate is not ''consistent'' in the probabilistic sense. Examples included the mean when the

population distribution does not have a finite variance and when the maximum or the minimum is taken from a sample. This is illustrated in Athreya (1987a,b), Knight (1989), Angus (1993), and Hall et al. (1993). The first published example of an inconsistent bootstrap estimate appeared in Bickel and Freedman (1981). Shao et al. (2000) showed that a particular approach to bootstrap estimation of individual bioequivalence is also inconsistent. They also provide a modification that is consistent. Generally, the bootstrap is consistent when the central limit theorem applies (a sufficient condition is Lyapanov's condition that requires existence of the $2 + \delta$ moment of the population distribution). Consistency results in the literature are based on the existence of Edgeworth expansions; so, additional smoothness conditions for the expansion to exist have also been assumed (but it is not known whether or not they are necessary).

One extension of the bootstrap called m-out of-n was suggested by Bickel and Ren (1996) in light of previous research on it, and it has been shown to be a method to overcome inconsistency of the bootstrap in several instances. In the m-out of-n bootstrap, sampling is with replacement from the original sample but with value of m that is smaller than n. See Bickel et al. (1997).

Some bootstrap approaches in time series have been shown to be inconsistent. Lahiri (2003) covered the use of bootstrap in time series and other dependent cases. He showed that there are remedies for the m-dependent and moving block bootstrap cases that are consistent.

4.2 Definition and relationship to the delta method and other resampling methods

We will first provide an informal definition of bootstrap to provide intuition and understanding before a more formal mathematical definition. The objective of bootstrapping is to estimate the distribution of a statistic based on the data, such as a mean, median, or standard deviation. We are also interested in the properties of the distribution for the parameter's estimate and may want to construct confidence intervals. But we do not want to make overly restrictive assumptions about the form of the distribution that the observed data came from.

For the simple case of independent observations coming from the same population distribution, the basic element for bootstrapping is the empirical distribution. The empirical distribution is just the discrete distribution that gives equal weight to each data point (i.e., it assigns probability $1/n$ to each of the original n observations and shall be denoted F_n).

Most of the common parameters that we consider are functionals of the unknown population distribution. A functional is simply a mapping that takes a function F into a real number. In our case, we are only interested in the functional of cumulative probability distribution functions. So, for example, the mean and the variance of a distribution can be represented as functional in the following way. Let μ be the mean

for a distribution function F, then $\mu = \int x\ dF(x)$. Let σ^2 be the variance then $\sigma^2 = \int (x-\mu)^2\ dF(x)$. These integrals over the entire possible set of x values in the domain of F are particular examples of functional. It is interesting that the sample estimates most commonly used for these parameters are the same functional applied to the F_n.

Now the idea of bootstrap is to use only what you know from the data and not introduce extraneous assumptions about the population distribution. The " bootstrap principle'' says that when F is the population distribution and $T(F)$ is the functional that defines the parameter, we wish to estimate based on a sample of size n, let F_n play the role of F and F_n^*, the empirical distribution function of the bootstrap sample (soon to be defined), play the role of F_n in the resampling process. Note that the original sample is a sample of n independent identically distributed observations from the distribution F and the sample estimate of the parameter is $T(F_n)$. So, in bootstrapping we let F_n play the role of F and take n independent and identically distributed observations from F_n. Since F_n is the empirical distribution, this is just sampling randomly with replacement from the original data.

Suppose we have n = 5 and the observations are X_1=7, X_2=5, X_3=3, X_4=9 and X_5=6 and that we are estimating the sample mean, (7 +5+ 3 + 9 + 6)/5 = 6.0. Then sampling from the data with replacement generates what we call a bootstrap sample.

The bootstrap sample is denoted X_1^*, X_2^*, X_3^*, X_4^*, and X_5^*. The distribution for sampling with replacement from F_n is called the bootstrap distribution, which we previously denoted by F_n^*. The bootstrap estimate is then T(F_n^*). So the bootstrap sample might be X_1^*=5, X_2^*=9, X_3^*=7, X_4^*=7, and X_5^*=5.

Note that, although it is possible to get the original sample back typically some values get repeated one or more times and consequently others get omitted. For his bootstrap sample, the bootstrap estimate of the mean is (5 + 9 + 7 + 7 + 5)/5 = 6.6. Note that the bootstrap estimate differs from the original sample estimate, 6.0. If we take another bootstrap sample, we may get another estimate that may be different from the previous one and the original sample. Assume for the second bootstrap sample we get in this case the observation equal to 9 repeated twice. Then, for this bootstrap sample, X_1^*=9, X_2^*=9, X_3^*=6, X_4^*=7, and X_5^*=5, and the bootstrap estimate for the mean is 7.2.

If we repeat this many times, we get a histogram of values for the mean, which we will call the Monte Carlo approximation to the bootstrap distribution. The average of all these values will be very close to 6.0 since the theoretical mean of the bootstrap distribution is the sample mean. But from the histogram (i.e., resampling distribution), we can also see the variability of these estimates and can use the histogram to estimate skewness, kurtosis, standard deviation and confidence intervals.

In theory, the exact bootstrap estimate of the distribution of the parameter-estimate could be calculated by averaging appropriately over all possible bootstrap samples,

and in this example for the mean, that value would be 6.0. As noted before, there can be n^n distinct bootstrap samples (taking account of the ordering of the observations), and so even for n = 10, this becomes very large (i.e., 10 billion). So, in practice, a Monte Carlo approximation is used.

If you randomly generate M = 10,000 or 100,000 bootstrap samples, the distribution of bootstrap estimates will approximate the bootstrap distribution for the estimate. The larger M is the closer the histogram approaches the true bootstrap distribution. Here is how the Monte Carlo approximation works:

1. Generate a sample with replacement from the empirical distribution for the data (this is a bootstrap sample).
2. Compute T(F_n^*) the bootstrap estimate of T(F). This is a replacement of the original sample with a bootstrap sample and the bootstrap estimate of T(F) in place of the sample estimate of T(F).
3. Repeat steps 1 and 2 M times where M is large, say 100,000.

Now a very important thing to remember is that with the Monte Carlo approximation to the bootstrap, there are two sources of error:

1. The Monte Carlo approximation to the bootstrap distribution, which can be made as small as you like by making M large;
2. The approximation of the bootstrap distribution $T(F_n^*) - T(F_n)$ to the distribution of $T(F_n) - T(F)$.

If $T(F_n^*) - T(F_n)$ converges as $n \to \infty$ to the same limit as the distribution of $T(F_n) - T(F)$, then bootstrapping works.

The probability theory associated with the bootstrap is beyond the scope of this text and can be found in books such as Hall (1992). What is important is that we know that consistency of bootstrap estimates has been demonstrated in many cases and examples where certain bootstrap estimates fail to be consistent are also known. There is a middle ground, which are cases where consistency has been neither proved nor disproved. In those cases, simulation studies can be used to confirm or deny the usefulness of the bootstrap estimate. Also, simulation studies can be used when the sample size is too small to count on asymptotic theory, and its use in small to moderate sample sizes needs to be evaluated.

But we need to mention that the above method is a non parametric bootstrap and there is also the method of parametric bootstrap which we are going to use in the following chapters. Whereas non parametric bootstraps make no assumptions about how your observations are distributed, and resample your original sample, parametric bootstraps resample a known distribution function, whose parameters are estimated from your sample. These bootstrap estimates are either used to attach confidence limits non parametrically-or a second and more parametric models are fitted using parameters

estimated from the distribution of the bootstrap estimates, from which for example, confidence limits are obtained analytically.

The advantages of this approach compared to the non parametric bootstrapping can be summarized as follows:

a) In non parametric bootstrap, samples are drawn from a discrete set of n observations. This can be a serious disadvantage in small sample sizes because spurious fine structure in the original sample, but absent from the population sampled, may be faithfully reproduced in the simulated data.
b) Another concern is that because small samples have only a few values, covering a restricted range, non parametric bootstrap samples underestimate the amount of variation in the population you originally sampled. As a result, statisticians generally see samples of 10 or less as too small for reliable non parametric bootstrapping.

4.2.1 Jackknife

The jackknife was introduced by Quenouille (1949). Quenouille's aim was to improve an estimate by correcting for its bias. Later on, Tukey (1958) popularized the method and found that a more important use of the jackknife was to estimate standard errors of an estimate. It was Tukey who coined the name jackknife because it was a statistical tool with many purposes. While bootstrapping uses the bootstrap samples to estimate variability, the jackknife uses what are called pseudovalues.

First consider an estimate $\tilde{u}$ based on a sample of size n of observations independently drawn from a common distribution F. Here, just as with the bootstrap, we again let F_n be the empirical distribution for this data set and assume that the parameter $u = T(F)$, a functional; $\tilde{u} = \mathrm{T}(F_n)$, $\tilde{u}_{(i)} = \mathrm{T}(F_{n(i)})$, where $F_{n(i)}$ is the empirical distribution function for the n-1 observations obtained by leaving the i-th observation out. If $\tilde{u}$ is the population variance, the jackknife estimate of variance of σ^2 is obtained as follows:

$$\sigma^2_{jack} = n\sum_{i=1}^{n}\frac{(\tilde{u}_{(i)}-u^*)^2}{n-1},$$

where $u^* = \sum_{i=1}^{n}\frac{\tilde{u}_{(i)}}{n}$. The jackknife estimate of standard error for $\tilde{u}$ is just the square root for σ^2_{jack}. Tukey defined the pseudovalue as $\tilde{u}_i = \tilde{u} + (n\text{-}1)(\tilde{u} - \tilde{u}_{(i)})$. Then the jackknife estimate of the parameter u is $u_{jack} = \sum_{i=1}^{n}\frac{\tilde{u}_i}{n}$. So the name pseudovalue comes about because the estimate is the average of pseudovalues. Expressing the estimate of the variance of the estimate $\tilde{u}$ in terms of the pseudovalues we get:

$$\sigma^2_{jack} = \sum_{i=1}^{n} \frac{(\tilde{u}_i - u_{jack})^2}{n(n-1)}.$$

In this form, we see that the variance is the usual estimate for variance of a sample mean. In this case, it is the sample mean of pseudovalues. Like the bootstrap, the jackknife has been a very useful tool in estimating variances for more complicated estimators such as trimmed or Winsorized means.

One of the great surprises about the bootstrap is that in cases like trimmed mean, the bootstrap does better than the jackknife (Efron, 1982, pp. 28-29). For the sample median, the bootstrap provides a consistent estimate of the variance but the jackknife does not! See Efron (1982, p. 16 and chapter 6). In that monograph, Efron also showed, using Theorem 6.1, that the jackknife estimate of standard error is essentially the bootstrap estimate with the parameter estimate replaced by a linear approximation of it. In this way, there is a close similarity between the two methods, and if the linear approximation is a good approximation, the jackknife and the bootstrap will both be consistent. However, there are complex estimators where this is not the case.

4.2.2 Delta method

It is often the case that we are interested in the moments of an estimator. In particular, for these various methods, the variance is the moment we are most interested in. To illustrate the delta method, let us define $\varphi = f(\alpha)$ where the parameters φ and α are both one-dimensional variables and f is a function differentiable with respect to α. So there exists a Taylor series expansion for f at a point α_0. Carrying it out only to first order, we get $\varphi = f(\alpha) = f(\alpha_0) + (\alpha\text{-}\alpha_0)f'(\alpha_0)$ + remainder terms and dropping the remainder terms leaves:

$$\varphi = f(\alpha) = f(\alpha_0) + (\alpha - \alpha_0)f'(\alpha_0)$$

or

$$f(\alpha) - f(\alpha_0) = (\alpha - \alpha_0)f'(\alpha_0).$$

Squaring both sides of the last equation gives us:

$$[f(\alpha) - f(\alpha_0)]^2 = (\alpha - \alpha_0)^2[f'(\alpha_0)]^2.$$

Now we want to think $\varphi = f(\alpha)$ as a random variable, and upon taking expectations of the random variables on each side of the equation, we get:

$$E[f(\alpha) - f(\alpha_0)]^2 = E(\alpha - \alpha_0)^2[f'(\alpha_0)]^2. \quad (1.1)$$

Here, α and $f(\alpha)$ are random variables, and α_0, f(α_0), and $f'(\alpha_0)$ are all constants. Equation 1.1 provides delta method approximation to the variance of $\varphi = f(\alpha)$ since the left-hand side is approximately the variance of φ and the right-hand side is the variance of α multiplied by the constant $[f'(\alpha_0)]^2$ if we choose α_0 to be the mean of α.

4.3 Deviance

Deviance is a quality of fit statistic for a model that is often used for statistical hypothesis testing. It is a generalization of the idea of using the sum of squares of residuals in ordinary least squares to cases where model-fitting is achieved by maximum likelihood.

The deviance for a model M_0, based on a data set $\boldsymbol{y}$, is defined as:

$D(\boldsymbol{y}) = -2(\log(p(\boldsymbol{y} \mid \hat{\theta}_0)) - \log(p(\boldsymbol{y} \mid \hat{\theta}_s)))$.

Here $\hat{\theta}_0$ denotes the estimated values of the parameters in model M_0, while $\hat{\theta}_s$ denotes the parameter estimates for the "full'' (or saturated) model. Both sets of parameter estimates are implicitly functions of the observations $\boldsymbol{y}$. Here the **saturated model** is a model with a parameter for every observation so that the data are fitted exactly. This expression is simply -2 times the log-likelihood ratio of the reduced model compared to the saturated model. The deviance is used to compare two models – in particular in the case of generalized linear models where it has a similar role to residual variance from AN.O.VA. in linear models (RSS).

Suppose in the framework of the generalized linear models, we have two nested models, M_1 and M_2. In particular, suppose that M_1 contains the parameters in M_2 and k additional parameters. Then under the null hypothesis that M_2 is the true model, the difference between the deviances for the two models follows an approximate chi-squared distribution with k degrees of freedom.

4.4 Testing goodness of fit in the fertility problem using the bootstrap

For the fertility problem, the log-likelihood for the saturated model minus the constant which does not affect the maximizations is:

$\log(p(\boldsymbol{y}|\theta_s)) = \sum_{i=1}^{13} n_i \log \pi_i$, where $\pi_i = \frac{n_i}{n}$, i=1,…,13.

For the non-smokers data, the saturated log-likelihood is:

$$log(p(\boldsymbol{y}|\theta_s)) = 198log(\tfrac{198}{486}) + 107log(\tfrac{107}{486}) + 55log(\tfrac{55}{486}) + 38log(\tfrac{38}{486}) + 18log(\tfrac{18}{486}) + 22log(\tfrac{22}{486}) + 7log(\tfrac{7}{486}) + 9log(\tfrac{9}{486}) + 5log(\tfrac{5}{486}) + 3log(\tfrac{3}{486}) + 6log(\tfrac{6}{486}) + 6log(\tfrac{6}{486}) + 12log(\tfrac{12}{486}) = -884.7072$$

For the geometric distribution, model M_1, the maximized multinomial log-likelihood, minus the same constant, is, from Chapter 3

$$log(p(\boldsymbol{y}|\theta_1)) = -907.2,$$

resulting in the observed deviance $D_1(\boldsymbol{y}) = 46.49125$.

We next use the parametric bootstrap to generate 500 samples from M_1 and calculate the deviance from each sample. We could then make a histogram to see where the observed deviance is located relative to the other deviances from the simulated samples. It is important to mention that the log- likelihoods are multinomial log-likelihoods.

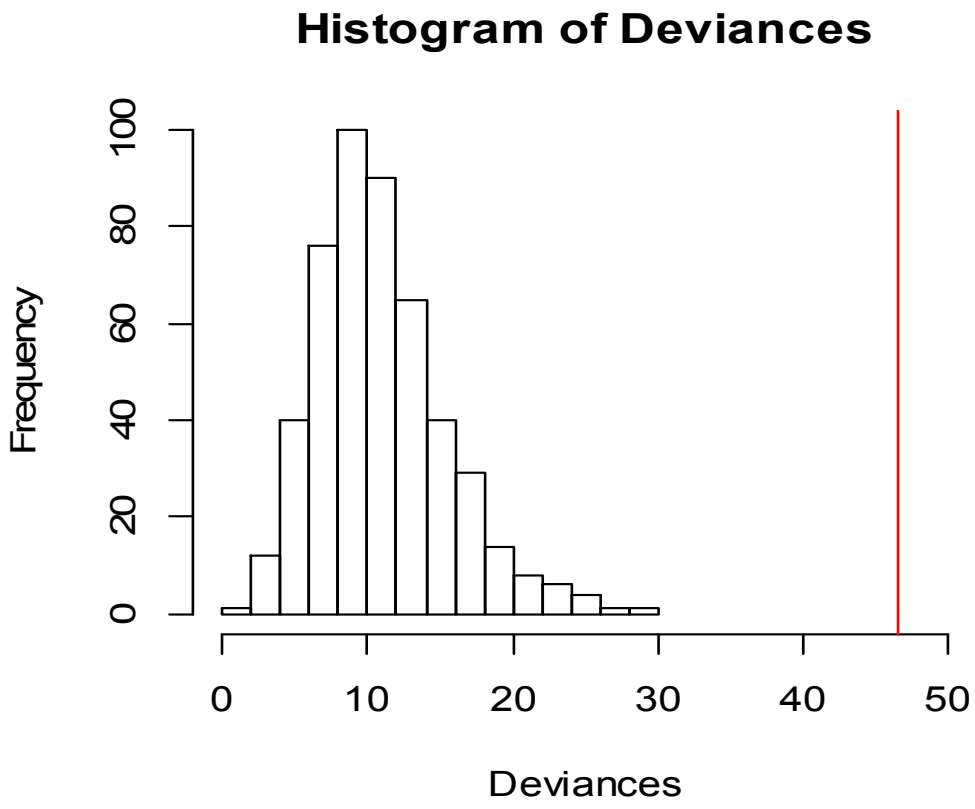

Figure 4.1: Histogram of simulated deviances from geometric distribution. The red line indicates the location of the observed deviance.

Fig.4.1 provides a histogram of the simulated deviances from the geometric distribution. The location of the observed deviance is indicated by the red line. We can see that the geometric model using multinomial log-likelihood does not fit the data well (p-value=0) because the red line is not inside the histogram, on the contrary it is far right. The real question is to find out how good is the beta-geometric model in contrast to the geometric model.

For the beta-geometric distribution (model M_2), the maximized multinomial log-likelihood is -890.3918, resulting in the observed deviance:

$D_2(\boldsymbol{y})$= -2(-890.3918-(-884.7072) = 11.36915

We then use parametric bootstrap sampling to check the fit of the beta-geometric distribution using 500 samples as above. Fig. 4.2 provides a histogram of the simulated deviances relative to the observed deviance, indicated by the red line.

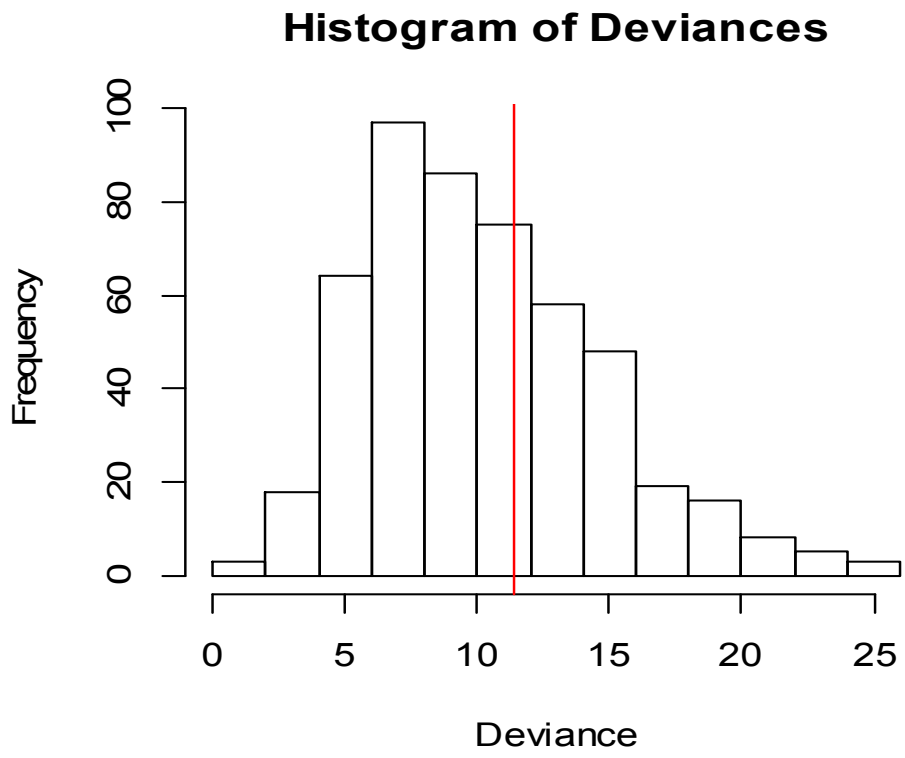

Figure 4.2: Histogram of the simulated deviances from beta-geometric distribution. The red line indicates the location of the observed deviance.

We can see that the red line is close enough to the centre of the histogram and we can say that the beta-geometric fits the observed data well (p-value=0.358). Thus the results from the bootstrap are consistent with the results from the chi-square test in Chapter 3, however the former procedure requires fitting the model many times while the latter is strictly speaking asymptotic. In the next chapter we consider a new method called calibrated simulation.

Chapter 5

CALIBRATED SIMULATION

5.1 Introduction

In the context of integrated population modeling in ecology, where potentially several data sets are being analyzed in combination, Besbeas and Morgan (2014) proposed a new approach, called calibrated simulation, for judging how well models fit data. Here comparative data sets are obtained from simulating data when model parameter values are obtained from the assumed asymptotic normal distribution of the maximum-likelihood estimators from the real data. The approach is motivated and justified by Bayesian p-values. It is attractively simple, as it limits the additional model-fitting that is required, and an appreciable improvement in efficiency is obtained relative to the bootstrap. Calibration of the resulting statistics is achieved as repeated data sets are easily simulated from the fitted model, and time-consuming multiple Markov chain Monte Carlo runs are not required. The method requires the specification of model discrepancy measures and the authors show how different measures can highlight different aspects of fit.

In this chapter we illustrate the use of calibrated simulation to examine how well the geometric and the beta-geometric distributions fit the women non-smokers data set. We also consider an unpublished modification of the method where simulated data sets are obtained from the maximum likelihood point estimates.

5.2 Calibrated simulation

The idea of calibrated simulation was suggested by Brooks et al (2000) in the context of the analysis of mark recovery and recapture data from wild birds, and it is also suggested by Johnson (2004). The approach is motivated by Bayesian p-values; see eg., Brooks et al (2000), where multiple simulations are obtained from posterior distribution for the parameters of the model being considered. The method was

proposed by Besbeas and Morgan in the context of integrated population modeling in ecology. Once the integrated model is fitted to all of the data then *s* simulated data sets, of dimensions matched to those of the real data sets, are obtained repeatedly from the component models, each one with parameter values obtained by simulating from the assumed asymptotic multivariate normal distribution of the maximum-likelihood parameter estimates from fitting the real data.

In detail, suppose that $\hat{\theta}$ and $\hat{\Sigma}$ are respectively the maximum-likelihood estimates from fitting the real data, and associated dispersion matrix obtained from inverting the observed information matrix evaluated at $\hat{\theta}$. For each simulated parameter value $\theta_i \sim N(\hat{\theta}, \hat{\Sigma})$ we might calculate a measure of the discrepancy between the data, $\boldsymbol{x}$ and the corresponding model, $D(\boldsymbol{x};\theta_i)$, and for each simulated parameter value θ_i we also simulate a new data set $\boldsymbol{x}_i$. For each new data set we then calculate $D(\boldsymbol{x_i}; \theta_i)$, and a scatter plot is obtained of $D(\boldsymbol{x_i}; \theta_i)$ vs $D(\boldsymbol{x};\theta_i)$. If the model fits the data well then one would expect approximately half of the points in the scatter plot to be above the line of unit slope through the origin as recommended by Brooks, S.P, Catchpole, E.A., and Morgan, B.J.T. (2000). The goodness of fit p-values we use are Fisherian p-values, i.e. probabilities of seeing something as weird or weirder than we actually saw. We denote the proportion of points above the line of unit slope by $p_c = n_c/s$ (we shall call those p_c's as p-values), where n_c is the corresponding number of points above the line. An attraction of this approach is that there is complete freedom in the choice of the measures of discrepancy that may be used, and furthermore more than one might be used for each data set, as recommended by Gelman et al (1996). For example, Millar and Meyer (2000) used four different measures when assessing the fit of a surplus – production model for fisheries data; one was a standard chi-square, while the other three were specific to the problem. They obtained p-values of 0.69, 0.27, 0.50, and 0.42 which they judged indicated that the model fitted the data sufficiently well. However we note the variation in the p-values obtained, which indicates the importance of taking several measures highlighting different aspects of fit. As observed by Johnson (2004), it also demonstrates that the distribution of p-values is unknown, and they cannot be calibrated. By running simulations for bootstrapped versions of the real data we provide such a calibration for the methods in this project, without the need for multiple Markov chain Monte Carlo simulations.

If uninformative prior distributions are assumed for the model parameters, and if the assumption of asymptotic normality for the distribution of maximum–likelihood estimators is justified then simulating as we do from the multivariate normal distribution will be equivalent to simulating from posterior distribution for the parameters, producing Bayesian p-values. It is therefore important to check the assumption of multivariate normality for the problems that we consider. Should the assumption of multivariate normality not hold then a possible approach, which we do not consider here, would be to sample from a kernel density estimate from additional bootstrap sampling.

5.4 Choice of discrepancy measure

5.4.1 Mark recovery data

For ring recovery data there are different discrepancy measures that may be used. Brooks et al (2000) use the Freeman–Tukey statistic (Freeman and Tukey, 1950) in which, for expected values $\{e_i\}$, we define the following discrepancy measure:

$$D_{FT}\ (\boldsymbol{x};\ \theta) = \sum_i (\sqrt{x_i} - \sqrt{e_i})^2,$$

and an alternative is the Pearson chi – square statistic, incorporating an amalgamation level m to accommodate small values. Details of these two measures and their asymptotic equivalence when the model is correct, are provided by Bishop et al (1975, p513). The difficulty with using the chi–square measure when data are sparse is the need for pooling cells with small expected values, which is not only arbitrary but results in differential weighting of the cells. If matching such extreme values is seen to be important then the chi–square discrepancy measure will indicate poor fit of the model. This explains how different discrepancy measures can lead to different values and indeed different conclusions. We therefore select the Freeman–Tukey measure for use in the work of this paper.

5.4.2 Census data

For any time series $\{y_t\}$ there are many alternative discrepancy measures that be used, based on the prediction errors, $\{y_t - \hat{y}_t\}$, where $\hat{y}_t$ are fitted values. Besbeas

and Morgan (2014) used two simple measures in their paper; these are the mean absolute percent prediction error (MAPE),

$$D_{MAPE}(\boldsymbol{y};\theta) = \frac{100}{n}\sum_{t=1}^{n}\left|\frac{(y_t - \widehat{y_t})}{y_t}\right|,$$

where n is the number of (non-missing) prediction errors, and the maximum percent error (MPE),

$$D_{MPE}(\boldsymbol{y};\theta) = 100\max\left\{\frac{(y_t - \widehat{y_t})}{y_t}\right\}.$$

In both cases the observations where $y_t = 0$ are ignored. In practice careful thought needs to be given to the selection of an appropriate discrepancy measure and there is a wide range of alternatives that may be appropriate in different applications.

5.5 Examining the fertility problem using the new method

In this section we consider the fertility problem assuming that the number of cycles to conception follows a geometric or a beta-geometric distribution and we shall examine the goodness of fit of both of these models using calibrated simulation. We focus on the non-smokers data, which are provided in Table 1.1.

The work of this chapter is based on the multinomial likelihood which is more appropriate for these data than the simple likelihood as the numbers of cycles >12 are amalgamated and from that likelihood we are going to compute the maximum likelihood estimator.

5.5.1 Goodness of fit testing that the data follow a geometric distribution using calibrated simulation

From Chapter 3, the maximum likelihood estimates from the geometric distribution are:

Estimate of p	**Value of log-likelihood**	**Hessian matrix**
0.3317 (0.0125)	**-907.2**	**6446.462**

Following Besbeas and Morgan (2014) we are going to use the Freeman-Tukey discrepancy measure given by:

$$D_{FT}(\boldsymbol{x};\theta) = \sum_i (\sqrt{x_i} - \sqrt{e_i})^2, \quad i = 1,2,3,\ldots \#\text{classes}$$

where x_i and e_i are the observed and expected frequencies, respectively.

We proceed by generating parameter values $\hat{p}_i$ from the asymptotic normal distribution of the maximum-likelihood estimator $\hat{p}$:

$$p_i \sim N(0.3317, 0.0125^2), i = 1,2,\ldots,500$$

Thus the first 150 of the simulated parameter values are:

0.3255	0.3358	0.3218	0.3451	0.3425
0.3276	0.3377	0.3443	0.349	0.3338
0.321	0.3371	0.3459	0.3114	0.3194
0.3194	0.338	0.3202	0.3176	0.3174
0.3408	0.3424	0.33	0.3205	0.3507
0.3469	0.3437	0.3298	0.3573	0.3316
0.3353	0.3234	0.3261	0.3209	0.3524
0.339	0.3294	0.3378	0.3451	0.3108
0.3411	0.3431	0.3516	0.2991	0.3365
0.3445	0.3397	0.3334	0.3447	0.3306
0.3232	0.3431	0.3301	0.3397	0.3264
0.3346	0.3252	0.318	0.3486	0.3261
0.3406	0.3281	0.3435	0.3336	0.3355
0.3335	0.34	0.3595	0.3403	0.3328
0.3156	0.3567	0.3362	0.337	0.3134
0.3235	0.3391	0.308	0.3378	0.3189
0.3216	0.3596	0.3284	0.3074	0.3194
0.3166	0.3282	0.3289	0.3119	0.3217
0.3107	0.3351	0.3435	0.3342	0.3274
0.3114	0.3365	0.3455	0.3406	0.3341
0.3312	0.3375	0.3358	0.3468	0.3361
0.3269	0.3342	0.3336	0.3266	0.337
0.3389	0.3388	0.3332	0.3462	0.3407
0.3421	0.3388	0.3083	0.3212	0.3179
0.3111	0.3189	0.3395	0.311	0.3119
0.3407	0.3386	0.3496	0.3201	0.3473
0.3326	0.3562	0.3369	0.332	0.3406
0.3446	0.3244	0.3019	0.3166	0.3225
0.3354	0.3125	0.3225	0.3337	0.3117

0.3322	0.33	0.3332	0.351	0.3427

For each simulated parameter value $\hat{p}_i$ we simulate data sets $\boldsymbol{x}_i$ from the model, of dimensions matched to the observed data, ie n=486. The first 7 out of a total of 500 simulated data sets are shown below:

		Simulated sets x_i using $\hat{p}_i$, i=1,…,7						
cycle	**Observed frequencies**	**1**	**2**	**3**	**4**	**5**	**6**	**7**
1	198	144	150	165	165	151	153	142
2	107	112	123	104	114	99	114	108
3	5	82	60	82	83	73	67	84
4	38	45	38	42	28	51	48	59
5	18	43	34	32	27	35	38	37
6	22	22	32	18	27	21	21	14
7	7	18	16	16	12	17	17	12
8	9	4	12	10	7	19	14	11
9	5	6	11	0	7	6	5	8
10	3	1	4	9	3	7	4	2
11	6	2	2	2	4	2	1	3
12	6	2	3	1	3	1	0	1
>12	12	5	1	5	6	4	4	5

We also calculate corresponding expected values under the model using $\hat{p}_i$ and we are going to have 500 sets of expected values different from each other. The expected frequencies corresponding to the 7 simulated data sets above are shown below:

	Expected frequencies e_i using $\hat{p}_i$, i=1,2,…..500						
cycle	**1**	**2**	**3**	**4**	**5**	**6**	**7**
1	166	154	169	158	157	151	162
2	109	105	110	107	106	104	108
3	72	72	72	72	72	72	72
4	47	49	47	49	49	49	48
5	31	33	31	33	33	34	32
6	21	23	20	22	22	23	21
7	14	16	13	15	15	16	14
8	9	11	8	10	10	11	9
9	6	7	6	7	7	8	6
10	4	5	4	5	5	5	4

11	3	3	2	3	3	4	3
12	2	2	2	2	2	3	2
>12	3	5	3	4	4	6	4

For each set of expected frequencies, we calculate a measure of the discrepancy between the observed data, $\boldsymbol{x}$, and the assumed model, $D(\boldsymbol{x}; p_i)$ using the Freeman-Tukey measure. Here are demonstrated the first 150 from the 500 calculated values:

$D(\boldsymbol{x}; \boldsymbol{p_i})$, $\boldsymbol{i}$=1,2,…,150				
10.912	11.25	11.013	11.173	11.075
11.911	11.008	11.267	10.885	11.516
10.874	11.123	11.151	10.877	11.409
11.412	10.875	10.997	11.057	11.25
11.543	10.921	12.41	11.797	11.081
12.499	10.952	11.263	11.041	11.062
11.065	11.142	11.129	10.874	10.878
10.874	11.431	11.844	11.601	11.153
11.824	10.967	10.892	11.449	11.552
11.073	11.072	12.9	10.879	10.882
11.344	10.89	11.065	12.131	11.875
11.252	10.879	11.333	12.188	11.07
11.876	10.896	10.908	11.528	10.998
12.92	10.985	11.043	11.839	11.986
11.058	11.147	11.748	12.394	11.552
10.897	11.042	10.908	11.436	10.938
10.875	12.983	12.643	11.991	11.036
10.911	11.381	11.191	12.108	10.883
11.173	11.446	11.103	11.32	10.909
13.12	10.973	10.984	10.917	11.496
11.813	12.26	11.866	11.189	10.982
11.201	10.95	11.383	10.89	10.917
10.874	11.135	11.53	11.685	11.123
11.057	11.596	11.685	11.398	10.875
12.264	11.319	11.012	10.874	11.039
11.705	10.874	11.772	11.429	10.904
12.468	10.912	12.999	13.519	10.924
11.469	11.133	11.715	11.6	12.541
12.896	10.91	11.79	13.742	11.276
10.938	11.389	10.924	11.701	11.215

Our next step is to calculate the Freeman–Tukey discrepancy measure to find the value between the simulated $\boldsymbol{x_i}$'s sets and the 500 different expected values and the result is 500 values. Here are demonstrated 150 from 500 values:

$D(x_i; p_i)$, i = 1,2,…,150				
45.774	56.132	26.379	46.162	42.372
4.931	15.519	15.242	90.544	35.003
79.713	13.759	34.651	16.638	14.722
47.867	22.101	51.217	17.547	2.548
2.316	47.599	37.862	16.667	13.296
45.282	35.459	43.378	45.243	4.06
35.498	56.339	20.999	11.781	33.346
3.044	29.227	41.317	34.652	41.007
22.318	19.357	42.392	29.881	40.547
43.209	32.541	62.169	2.352	56.764
51.154	15.671	29.955	51.921	32.056
49.798	2.808	4.718	29.648	13.595
47.318	88.489	4.356	31.883	31.605
2.011	20.053	38.546	58.932	45.249
2.619	45.595	22.489	41.819	47.904
62.649	2.098	22.084	3.209	29.908
23.024	10.931	36.805	10.498	60.593
50.173	13.877	22.644	16.515	49.716
3.939	11.366	38.891	34.171	23.607
21.639	28.689	27.665	47.797	29.399
1.11	32.443	27.757	38.538	31.783
4.283	30.009	46.664	27.123	67.773
60.835	24.878	23.575	27.489	60.854
13.056	28.553	23.022	29.261	18.544
12.087	1.089	2.795	80.704	17.983
22.897	17.216	32.517	11.981	23.506
43.124	30.273	22.892	16.231	24.206
35.386	63.357	26.585	32.343	3.514
25.729	39.854	74.187	44.709	17.287
51.408	33.284	41.815	3.497	27.374

We need to mention that the discrepancy values (observed $(D(\boldsymbol{x}; \hat{p_i}))$, expected $(D(\boldsymbol{x_i}; \hat{p}_i))$ are in matched pairs e.g. 10.912 is matched to 45.774 as we shall see in a plot later.

Our step move is to find a p-value. This is necessary in order to check the fit of the model. It is calculated by how many of these values derived from the simulated data are greater from those derived from the observed data and divided by their number which is 500. The p-value (p-value = $\frac{discrepancy\ values\ from\ simulated\ \ data > discrepancy\ value\ from\ the\ observed\ data}{number\ of\ the\ simulated\ data\ sets}$) is 0.002 and then we have the following plot:

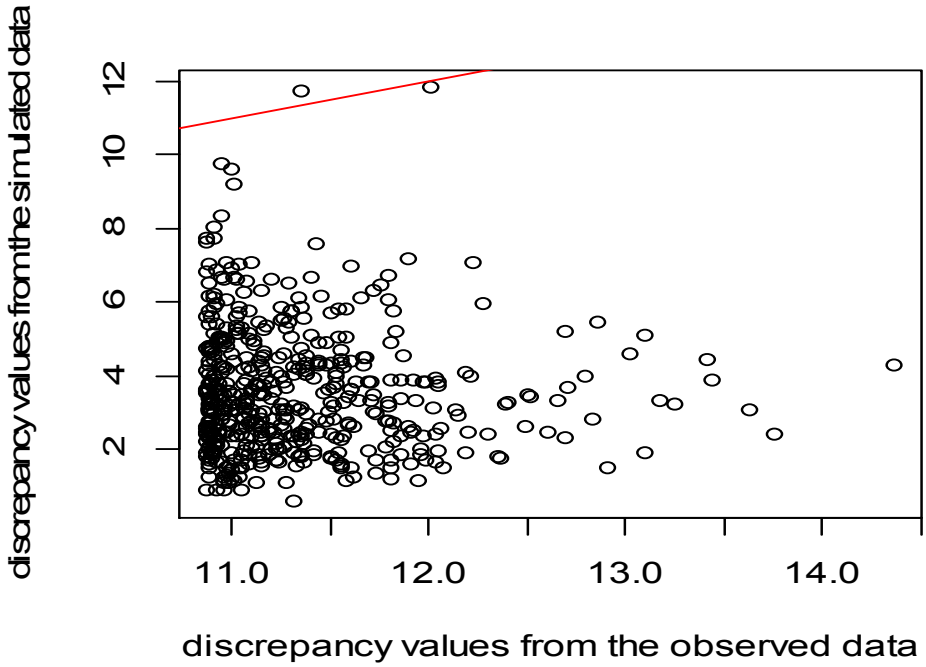

Figure 5.1: Scatter plot of D($\boldsymbol{x}$;$\hat{p}_i$) vs D($\boldsymbol{x}_i$;$\hat{p}_i$)

Fig.5.1 provides a scatter plot of $D(\boldsymbol{x};\hat{p}_i)$ vs $D(\boldsymbol{x}_i;\hat{p}_i)$. The proportion of points above the diagonal is the p-value of this method which is $0.002 < 0.05$ and therefore the geometric distribution does not fit well. Also the plot of the empirical concentrated distribution function of the p-values is:

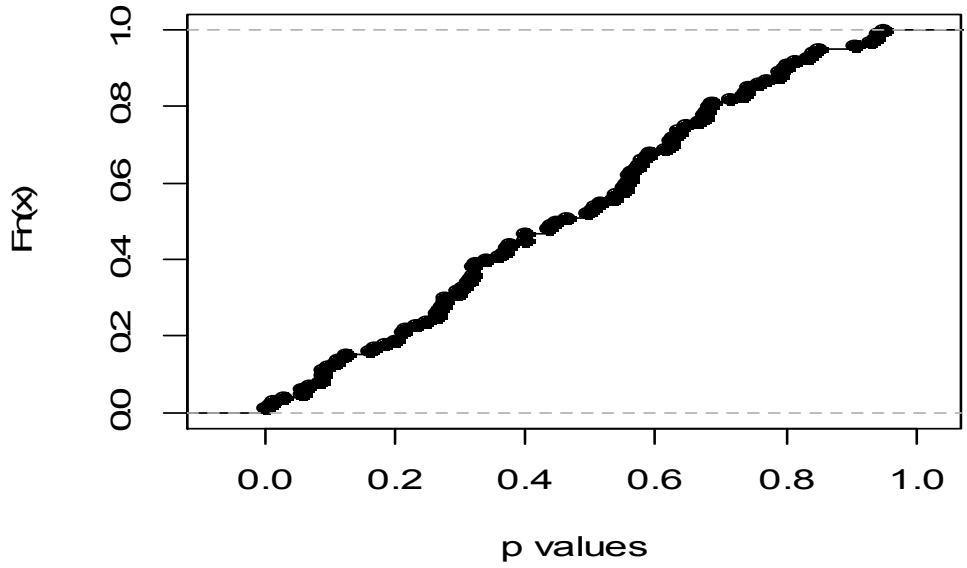

Figure 5.2: Plot of the p-values's e.c.d.f.

5.5.2 Calibrated simulation using the maximum-likelihood estimate

In this section, we consider a new variant of the method where the simulated data sets $\boldsymbol{x}_i$ i=1,…,500, are obtained using the maximum-likelihood point estimate $\hat{p}$ as opposed to random values from the asymptotic normal distribution.

The simulated data sets $\boldsymbol{x}_i$ from $\hat{p}$ are not very different from the data sets $\boldsymbol{x}_i$ above as the precision of the maximum-likelihood estimator is high. For comparison the first 7 of these data sets are presented below:

		Simulated sets x_i using $\widehat{\boldsymbol{p}}$						
Cycle	**Observed frequencies**	**1**	**2**	**3**	**4**	**5**	**6**	**7**
1	198	163	161	161	149	163	164	138
2	107	102	116	98	114	115	92	107
3	55	71	83	72	76	69	66	90
4	38	49	47	57	51	57	59	59
5	18	41	28	31	27	25	39	29
6	22	20	17	25	23	19	20	27
7	7	16	10	13	18	14	17	12
8	9	4	9	10	7	8	12	6

9	5	7	6	8	7	8	6	4
10	3	1	2	1	1	2	7	5
11	6	3	3	2	3	2	3	5
12	6	1	1	6	7	1	1	1
>12	12	8	3	2	3	3	0	3

The expected frequencies under the geometric distribution are given by:

Cycle	Expected frequencies using $\hat{p}$
1	161
2	108
3	72
4	48
5	32
6	21
7	14
8	10
9	6
10	4
11	3
12	2
>12	4

which are the same for each simulated data set $\boldsymbol{x}_i$ as a result of all depending on $\hat{p}$. The Freeman–Tukey discrepancy measure between the observed data and the model $D(\boldsymbol{x};\hat{p}) = 11.15178$. We are going to use that value 500 times as we shall see below.

As above we also calculate the Freeman–Tukey discrepancy measure between the simulated data set $\boldsymbol{x}_i$ and the model $D(\boldsymbol{x}_i;\hat{p})$ and we provide the first 150 of the 500 values below:

$D(\boldsymbol{x}_i;\hat{p})$, i=1,2…,150				
3.959	1.453	3.84	2.189	2.129
2	2.995	2.08	3	2.688
3.537	3.081	3.128	2.877	3.405
3.892	1.947	5.609	0.613	3.164
1.929	4.323	3.073	3.102	1.949
6.263	5.784	5.271	2.44	1.667
4.19	3.544	2.045	1.136	3.72

4.053	2.35	3.353	6.752	4.653
2.451	6.712	2.583	1.093	4.121
2.147	3.818	3.652	3.088	3.498
2.505	5.074	2.907	2.384	5.643
1.565	3.278	6.507	2.77	2.88
7.483	3.028	2.248	1.583	4.984
3.532	4.558	3.265	1.645	2.796
5.036	2.769	4.541	4.948	4.085
1.952	5.239	2.49	2.463	3.249
2.449	3.744	2.942	5.152	4.083
2.865	1.808	1.687	2.79	4.467
1.462	2.545	4.579	2.11	4.221
2.971	7.341	3.986	1.149	6.58
7.506	4.252	4.392	7.215	8.303
3.153	3.137	3.663	2.396	2.415
1.606	2.347	5.492	1.265	3.757
2.878	1.878	3.438	1.885	2.396
5.862	2.114	2.165	3.314	1.276
1.833	2.54	2.585	1.62	1.672
2.086	3.359	3.872	4.489	3.896
3.898	1.856	5.34	3.299	1.452
2.922	3.257	3.185	5.893	3.647
1.682	1.852	4.283	3.631	2.24

It is obvious that the values are different so our next step is to find a p-value. As above this p-value is calculated by how many of the values derived from the simulated data are greater from the one derived from the observed data and we are going to divide their sum by their number which is 500. The p-value (p-value = $\frac{discrepancy\ values\ from\ simulated\ \ data > discrepancy\ value\ from\ the\ observed\ data}{number\ of\ the\ simulated\ data\ sets}$) is 0.004 and then we are going to see the follow plot.

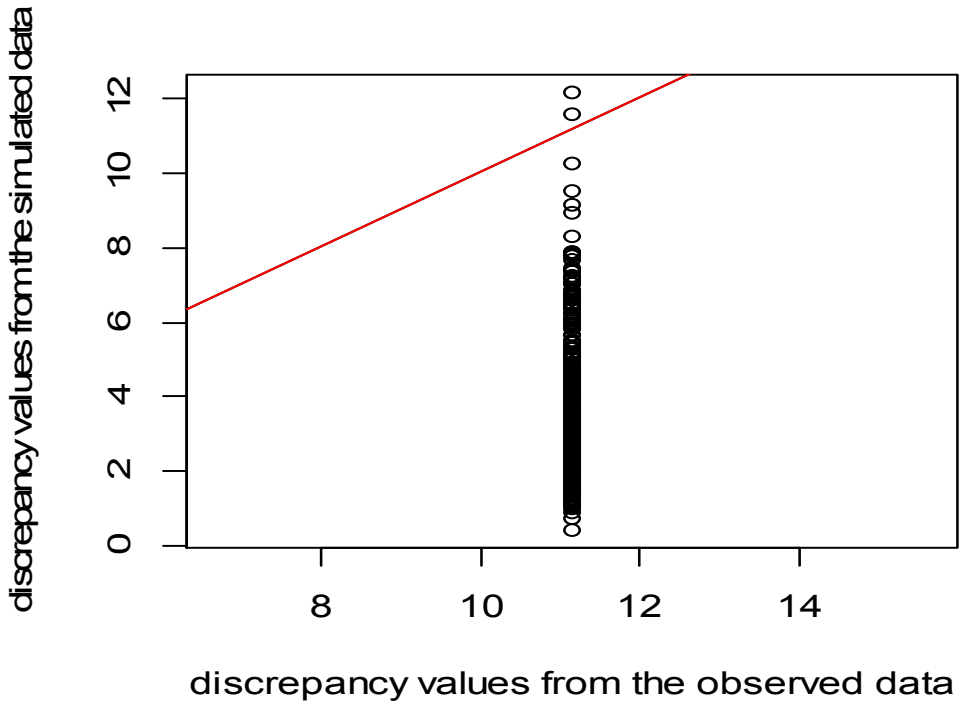

Figure 5.3: Scatter plot of D($\boldsymbol{x}$;$\hat{p}$) vs D($\boldsymbol{x}_i$: $\hat{p}_i$)

It is obvious that the points lie on a straight line since we use the M.L.E. The p-value of the method is thus 0.004<0.05 and therefore we also conclude that the geometric distribution does not fit the data well in line with previous methods. Also the plot of the empirical concentrated distribution function of the p-values is:

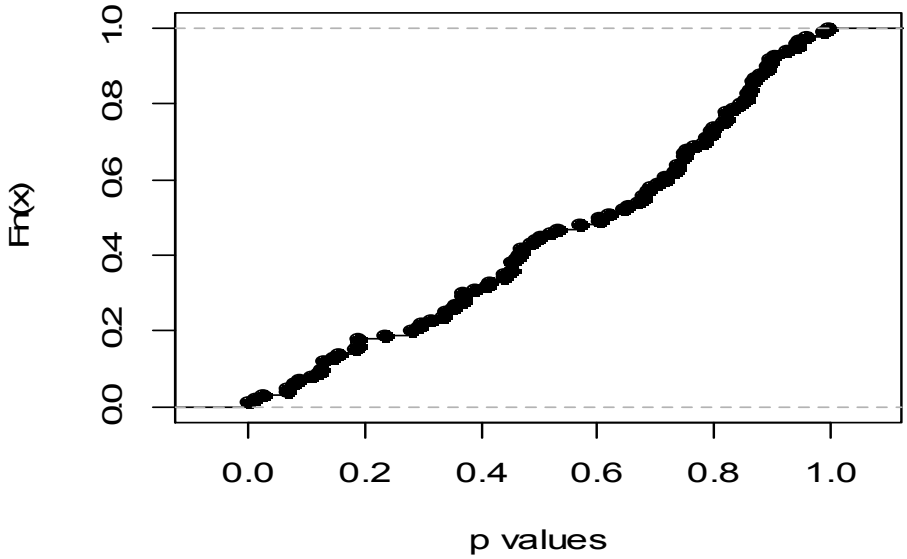

Figure 5.4: Plot of the p-values's e.c.d.f.

5.5.3 Sampling distribution of p-value

In general, the sampling distribution of the p-value of the method is unknown and Besbeas and Morgan (2014) propose the use of further simulation to calibrate the observed p-value. We describe and illustrate this procedure below.

We are going to generate 100 simulated "observed" data sets from which we are going to find 100 p-values for each case. The 100 simulated "observed" data sets are going to be generated using the M.L.E. from the observed data and the geometric distribution and we are going to calibrate the p-value from the original approach as well as the new variant. First let's calibrate the p-value from the variant, where the $\boldsymbol{x}_i$'s are generated from the M.L.E.

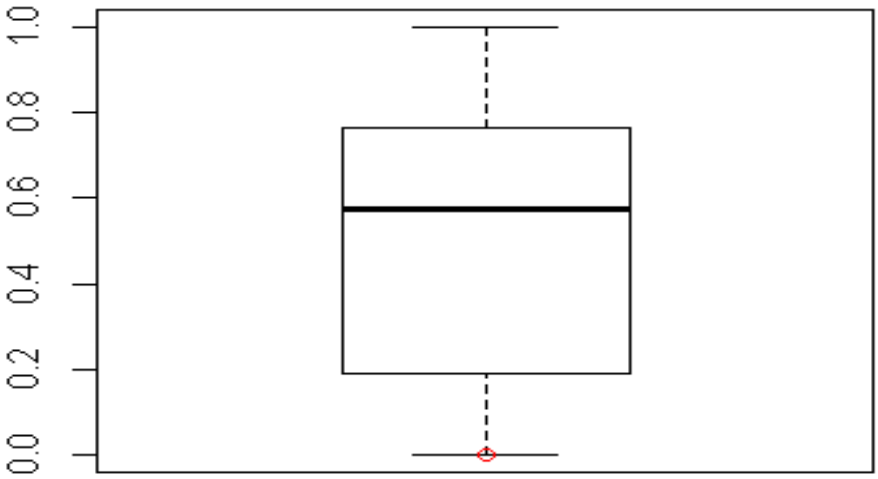

Figure 5.5: Boxplot of the simulated p-values. The circle indicates the location from the observed data

We can see that the sampling distribution of the p-value under the null is nearly uniform, and that the observed p-value (0.004) is far in the bottom of the boxplot. The observed p-value is therefore extreme and we can say that the geometric model is not suitable for the data. We repeat the procedure for the original approach where the $\boldsymbol{x}_i$ are generated from the $\hat{p}_i$ which in turn are generated from the asymptotic normal distribution of the maximum-likelihood estimator from each "observed" data set.

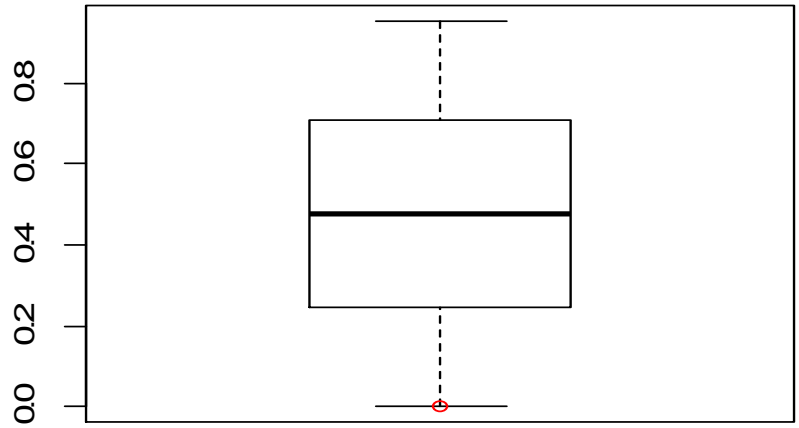

Figure 5.6:Boxplot of the p-values from the original approach. The circle indicates the location from the observed data.

We can see that the observed p-value (0.002) is also far in the bottom of the box-plot and so we can say here too that the geometric model is not appropriate for the original set of data.

5.5.4 Goodness of fit testing that the data follow a beta-geometric distribution using calibrated simulation

We now examine the goodness-of-fit of the beta-geometric distribution. Recall that the maximum likelihood estimates of the parameters based on the multinomial likelihood are as follows:

Estimate of α	Estimate of β	Value of log-likelihood
2.986042 (0.630)	**4.328728 (1.1359)**	**-890.3918**

and the hessian matrix is:

$$\mathbf{H}(\hat{\theta}) = \begin{bmatrix} 34.6496 & -18.51060 \\ -18.51060 & 10.66369 \end{bmatrix}$$

As in Section 5.5.1 we are going to use the Freeman-Tukey discrepancy measure given by:

$$D_{FT}\ (\boldsymbol{x};\theta) = \sum_i (\sqrt{x_i} - \sqrt{e_i})^2.$$

where x_i and e_i are the observed and the expected frequencies respectively.

We proceed by generating values $\hat{\theta}_i=(\hat{\alpha}_i,\hat{\beta}_i)$ from the asymptotic multivariate normal distribution of the maximum-likelihood estimator $\hat{\theta}$. Thus

$$\begin{bmatrix}\hat{\alpha}_i \\ \hat{\beta}_i\end{bmatrix} \sim N\left(\begin{bmatrix}2.986 \\ 4.328\end{bmatrix}, \begin{bmatrix}0.3971559 & 0.6894045 \\ 0.6894045 & 1.2904816\end{bmatrix}\right),\ i=1,2,\ldots,500$$

and the first 24 sets of simulated parameter are shown below:

$\boldsymbol{i}$	$\boldsymbol{\hat{\alpha}_i}$	$\boldsymbol{\hat{\beta}_i}$
1	3.022	3.269
2	2.904	6.315
2	3.209	4.404
3	2.737	4.817
4	2.193	4.401
5	2.587	6.75
6	2.757	4.11
7	3.088	3.515
8	3.449	2.582
9	3.805	4.384
10	4.02	5.738
11	3.311	4.197
12	2.545	4.557
13	2.812	3.622
14	4.18	3.313
15	2.747	3.157
16	1.784	2.972
17	2.458	4.775
18	2.597	3.941
19	2.394	4.771
20	3.501	2.282
21	2.712	2.781
22	4.258	3.346
23	3.125	3.049
24	2.857	6.202

For each pair of simulated parameters $(\hat{\alpha}_i,\hat{\beta}_i)$ we simulate data sets $\boldsymbol{x}_i$ from the model, of dimensions matched to the observed data, ie n = 486. The first 7 out of a total of 500 simulated data sets are shown below:

		Simulated data set x_i using $\hat{\alpha}_i, \hat{\beta}_i$ i=1,2,…,7						
Cycle	**Observed frequencies**	**1**	**2**	**3**	**4**	**5**	**6**	**7**
1	198	169	133	276	155	188	189	196
2	107	95	111	87	100	103	96	102
3	55	68	71	50	75	56	56	71
4	38	54	56	27	29	33	37	31
5	18	27	25	14	34	17	24	26
6	22	15	11	7	18	23	16	18
7	7	13	16	9	10	18	12	9
8	9	9	14	8	10	12	10	6
9	5	8	10	0	9	6	4	4
10	3	3	7	1	11	1	6	3
11	6	4	4	1	8	7	5	3
12	6	5	3	2	3	2	5	4
>12	12	16	25	4	24	20	26	13

We also calculate corresponding expected values under the model using the $\hat{\alpha}_i$ and $\hat{\beta}_i$, and we are going to have 500 sets of expected values different from each other in comparison with what we were doing in the previous section when we were using the M.L.E from each data set from the above 500. The expected frequencies corresponding to the 7 *simulated* data sets above are shown below:

	Expected frequencies e_i using $\hat{\alpha}_i$ and $\hat{\beta}_i$, i=1,…,7						
Cycle	**1**	**2**	**3**	**4**	**5**	**6**	**7**
1	168.152	160.75	278.261	163.364	182.321	190.476	199.082
2	99.623	97.187	97.897	94.912	98.328	98.281	101.304
3	62.515	62.143	43.981	59.557	58.484	56.968	57.585
4	41.073	41.55	22.925	39.587	37.332	35.805	35.422
5	28.02	28.812	13.218	27.519	25.141	23.889	23.12
6	19.723	20.595	8.201	19.829	17.657	16.688	15.805
7	14.257	15.104	5.38	14.713	12.828	12.092	11.213
8	10.544	11.323	3.689	11.188	9.583	9.025	8.201
9	7.953	8.652	2.621	8.686	7.329	6.905	6.153

10	6.104	6.722	1.917	6.865	5.717	5.394	4.717
11	4.757	5.3	1.438	5.511	4.537	4.289	3.684
12	3.759	4.234	1.101	4.484	3.655	3.464	2.924
>12	19.519	23.627	5.371	29.784	23.086	22.725	16.791

For each set of expected frequencies, we calculate a measure of the discrepancy between the observed data, $\boldsymbol{x}_i$, and the assumed model using the Freeman-Tukey measure.

$D[x;(\widehat{\alpha}_i,\widehat{\beta}_i)]$ i=1,2,…,100				
2.959	3.018	2.928	2.82	2.986
3.495	3.202	3.427	3.728	3.56
3.113	4.14	2.91	2.82	4.398
3.447	2.883	3.468	3.513	3.087
3.028	4.731	2.767	3.874	7.959
2.796	2.749	3.67	2.843	3.748
3.347	3.524	2.807	3.23	3.153
3.232	3.068	4.04	3.113	4.788
2.868	3.234	2.816	3.821	3.009
3.027	2.886	2.781	3.84	3.525
3.059	3.048	2.856	3.584	3.141
3.186	3.359	2.969	3.222	3.167
3.612	3.522	2.842	2.833	3.46
3.06	3.537	4.754	3.21	2.839
3.516	3.175	2.768	3.813	2.739
6.789	2.77	3.224	2.916	2.989
2.745	2.909	2.95	2.904	5.827
3.602	3.725	2.936	3.185	3.227
2.786	3.053	2.983	3.016	29.815
3.021	2.894	3.076	3.511	3.197

We also calculate the Freeman-Tukey discrepancy measure between the simulated data x_i and the assumed model, $D(\boldsymbol{x}_i;(\hat{\alpha}_i,\hat{\beta}_i))$ The first 145 discrepancy values $D(\boldsymbol{x};(\hat{\alpha}_i,\hat{\beta}_i))$ and $D(\boldsymbol{x}_i;(\hat{\alpha}_i,\hat{\beta}_i))$are shown in each case:

$D[\boldsymbol{x}_i;(\widehat{\alpha}_i,\widehat{\beta}_i)]$ i=1,2,…,145				
4.024	4.954	4.705	1.794	3.125
4.086	2.873	1.783	1.715	3.706
3.606	3.085	5.786	2.139	1.653
3.885	4.389	3.138	4.776	2.798
2.785	2.298	2.656	4.07	1.811
1.802	2.108	5.042	1.356	3.679
2.156	2.487	1.504	3.635	3.343

2.48	2.033	2.409	1.751	2.146
2.188	2.043	7.507	1.33	6.743
2.989	4.126	6.18	2.215	4.078
3.472	1.144	4.897	2.316	4.937
3.951	2.289	4.379	3.648	4.067
3.941	4.835	2.885	5.39	3.317
3.362	7.493	2.597	2.442	2.344
3.127	3.03	2.114	1.864	4.145
2.048	3.455	3.065	2.176	1.98
1.67	2.462	3.687	1.768	1.472
2.895	2.814	6.381	3.534	5.001
3.057	6.085	2.083	3.137	7.423
3.892	4.3	4.008	1.391	1.816
5.177	2.682	4.392	4.658	4.492
3.695	2.572	6.003	1.957	1.801
3.469	5.172	2.159	1.874	1.911
2.581	4.874	3.795	4.536	2.682
5.185	4.279	2.657	3.229	2.403
3.787	3.403	2.415	6.883	3.562
1.946	4.98	3.782	2.861	2.41
4.528	3.525	4.154	2.311	3.939
1.891	3.905	7.724	2.715	3.302

Note that these discrepancy measures are in matched pairs e.g.: 2.959 is matched to 4.024 etc. Fig 5.5 provides a scatter plot of $D(\boldsymbol{x};(\hat{\alpha}_i,\hat{\beta}_i))$ vs $D(\boldsymbol{x}_i;(\hat{\alpha}_i,\hat{\beta}_i))$. The proportion of points above the diagonal, which is the p-value of the method is 0.176>0.05 and therefore the beta-geometric distribution appear to fit the data well.

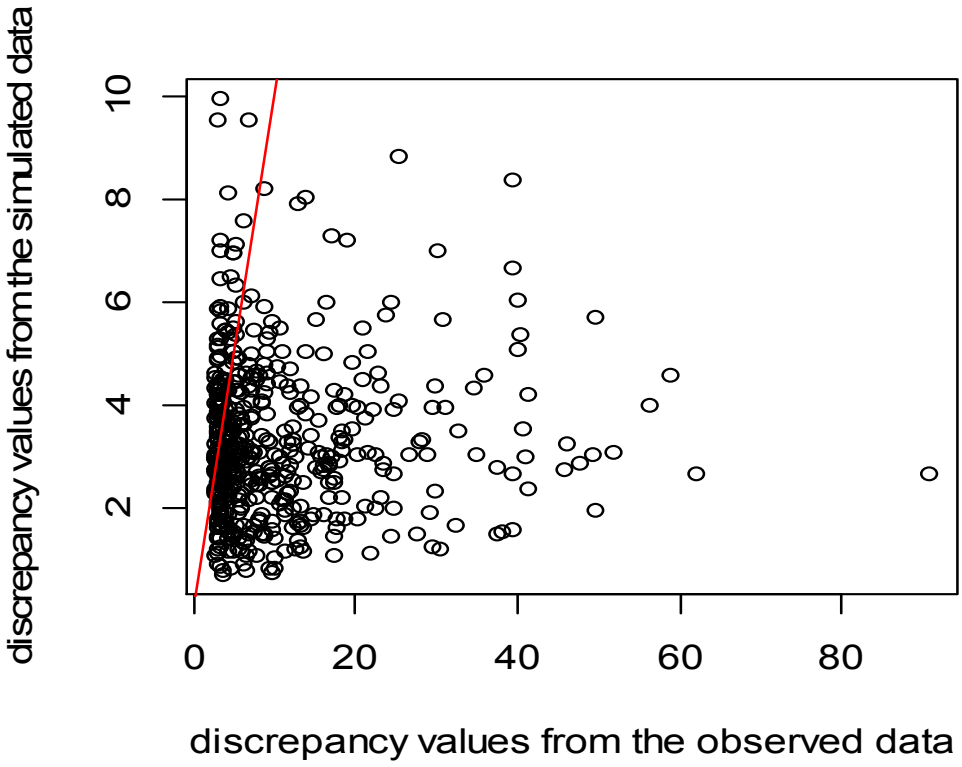

Figure 5.7: Scatter plot of D(x_i;($\hat{\alpha}_i$,$\hat{\beta}_i$)) vs D(x_i;($\hat{\alpha}_i$,$\hat{\beta}_i$))

Also the plot of the empirical concentrated distribution function of the p-values is:

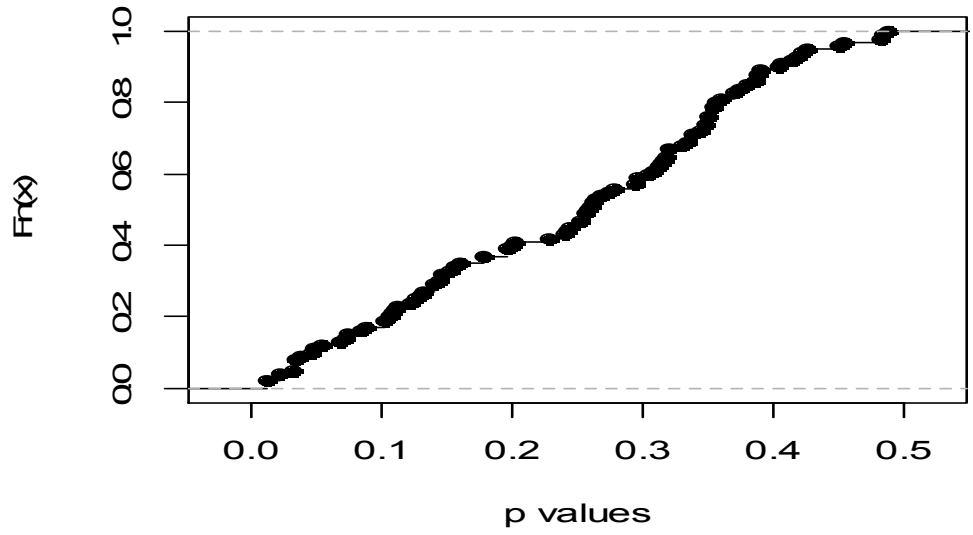

Figure 5.8: Plot of the p-values's e.c.d.f.

5.5.5 Calibrated simulation using the maximum-likelihood point estimates

In this section we consider a new variant of the method where the simulated data sets x_i, $i = 1, \ldots 500$ obtained using the maximum likelihood point estimates $(\hat{\alpha},\hat{\beta})$ as opposed to random values from the asymptotic normal distribution. The simulated data sets $\boldsymbol{x}_i$ are not very different from the data sets $\boldsymbol{x}_i$ above as precision of the maximum-likelihood estimator is high. For comparison the first 7 of these data sets are presented below:

		Simulated sets x_i-using $(\hat{\alpha},\hat{\beta})$						
Cycle	**Observed frquencies**	**1**	**2**	**3**	**4**	**5**	**6**	**7**
1	198	186	196	187	198	181	199	194
2	107	102	113	106	94	101	100	128
3	55	66	69	66	68	68	60	46
4	38	36	33	38	35	39	45	29
5	18	26	19	26	24	13	23	19
6	22	28	17	12	17	17	23	14
7	7	7	3	12	11	15	14	11
8	9	7	9	11	9	5	7	12
9	5	2	6	6	7	10	1	6
10	3	6	1	8	4	6	5	7
11	6	2	2	1	1	8	0	5
12	6	2	2	2	1	4	1	0
>12	12	16	16	11	17	19	8	15

The expected frequencies under the beta-geometric distribution are given by:

cycle	**Expected frequencies using $(\hat{\alpha},\hat{\beta})$**
1	198.395
2	103.286
3	59.087
4	36.253
5	23.482
6	15.881
7	11.127
8	8.029
9	5.939
10	4.488
11	3.455
12	2.703
>12	13.875

Which are the same for each simulated data set $\boldsymbol{x}_i$ as a result of all depending on $(\hat{a},\hat{b})$. The Freeman–Tukey discrepancy measure between the observed data and the model is : $D(\boldsymbol{x};(\hat{\alpha},\hat{\beta}))$ = 2.747282 and we are going to use that value 500 times as we shall see below.

As above we also calculate the Freeman–Tukey discrepancy measure between the simulated data set x_i and the model, $D(\boldsymbol{x}_i;(\hat{\alpha},\hat{\beta}))$ and we provide 150 example values of 500:

$D(\boldsymbol{x}_i;(\hat{\alpha},\hat{\beta}))$ i=1,…150				
4.159	3.115	1.568	1.887	2.461
5.119	3.885	5.368	4.448	1.581
2.444	5.822	2.403	5.367	4.15
1.961	3.495	3.981	3.495	1.172
5.082	7.217	3.99	1.951	6.321
8.114	1.841	1.934	2.685	2.63
6.411	3.295	2.053	1.286	4.444
3.532	2.354	9.571	1.44	2.582
4.338	2.95	5.985	2.467	1.05
1.677	2.001	1.788	1.853	3.974
3.772	5.725	4.914	3.263	3.524
5.613	3.877	2.942	2.183	7.476
3.077	4.159	6.742	2.287	4.319
2.597	2.976	3.817	5.984	3.041
1.151	2.224	2.634	2.572	1.725
1.361	3.688	1.978	2.686	3.278
2.304	2.37	1.955	2.984	3.348
2.086	3.094	3.596	5.12	2.907
1.416	3.169	1.957	3.258	4.08
3.083	2.542	1.419	3.878	4.031
3.777	1.028	4.483	5.951	3.169
2.701	5.053	3.139	3.988	4.042
3.473	6.522	1.897	4.102	1.976
5.021	2.545	2.931	4.99	2.882
1.531	1.379	9.921	7.224	5.341
3.254	2.707	5.022	4.718	3.641
3.759	3.373	1.356	3.196	3.342
2.731	2.693	1.461	1.853	1.63
4.135	3.825	4.411	2.259	5.511
1.509	3.786	3.551	2.12	2.217

1.068	2.673	3.028	3.253	9.438
2.184	2.561	3.516	2.665	4.565
3.793	2.321	3.055	5.106	1.109
2.389	1.576	3.261	3.938	2.575
3.731	1.618	8.388	2.055	0.91
6.365	5.887	1.355	5.3	3.077
3.742	1.742	2.27	4.318	1.887

Fig. 5.6 provides a scatter-plot of $D(\boldsymbol{x};(\hat{\alpha},\hat{\beta}))$ vs $D(\boldsymbol{x}_i;(\hat{\alpha},\hat{\beta}))$. The proportion of points above the diagonal which is the p-value of the method, is 0.588>0.05 and therefore the conclusion from the new variant is in agreement with that from the original method in Section 5.5.2.

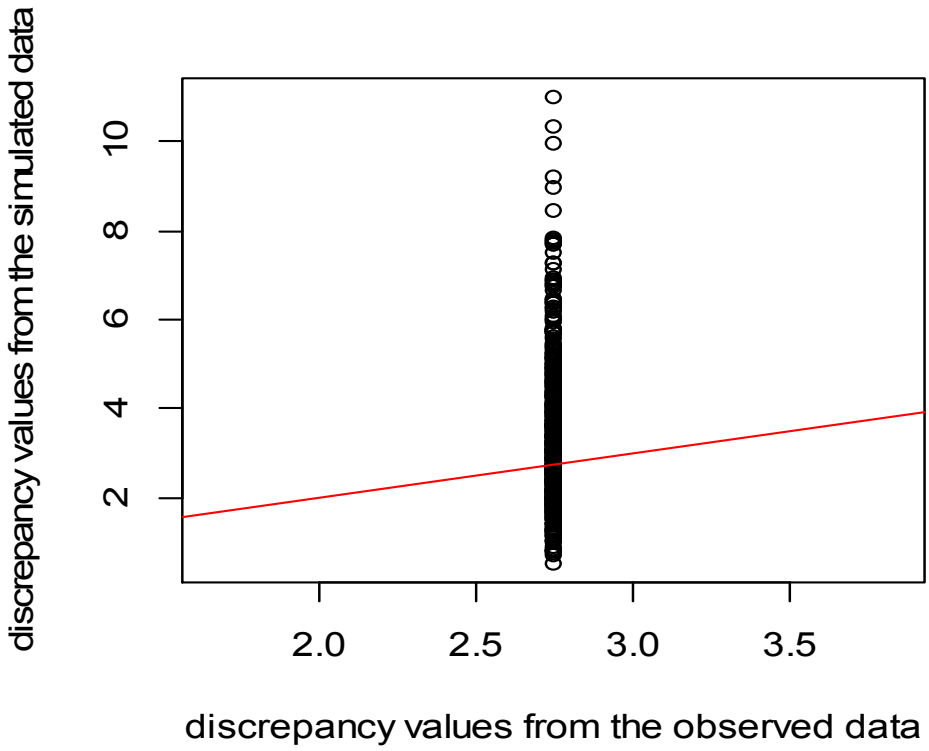

Figure 5.9: Scatter plot $D(\boldsymbol{x}_i;(\hat{\alpha},\hat{\beta}))$vs $D(\boldsymbol{x};(\hat{\alpha},\hat{\beta}))$

Also the plot of the empirical concentrated distribution function of the p-values is:

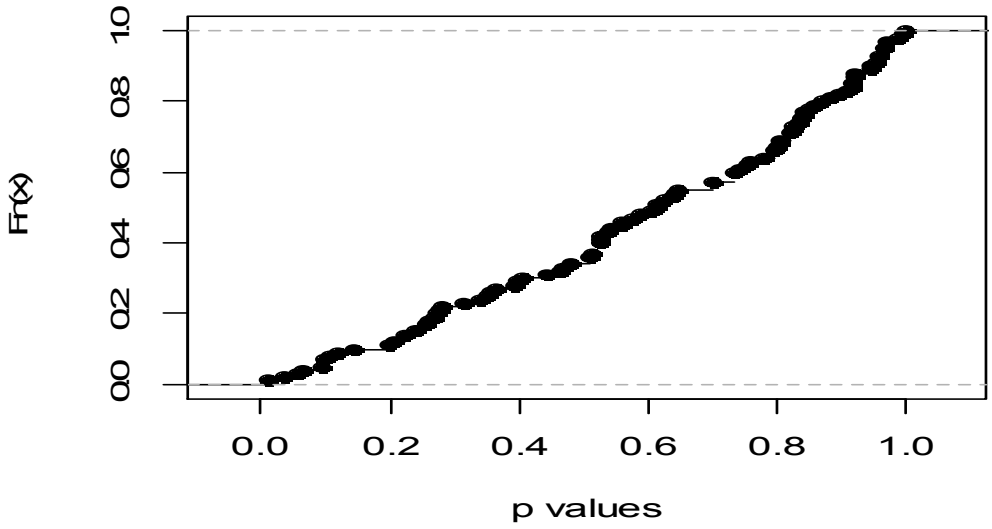

Figure 5.10: Plot of the p-values's e.c.d.f.

5.5.6 Sampling distribution of p-value

In general, the sampling distribution of the p-value of the method is unknown, and Besbeas and Morgan (2014) propose the use of further simulation to calibrate the observed p-value. We describe and illustrate this procedure below.

We are going to generate 100 simulated "observed" data sets from which we are going to find 100 p-values for each case. The 100 simulated "observed" data sets are going to be generated using the M.L.E. from the observed data and the beta-geometric distribution and we are going to calibrate the p-value from the original approach as well the new variant. First let's calibrate the p-value from the variant, where the x_i's are generated from the M.L.E.

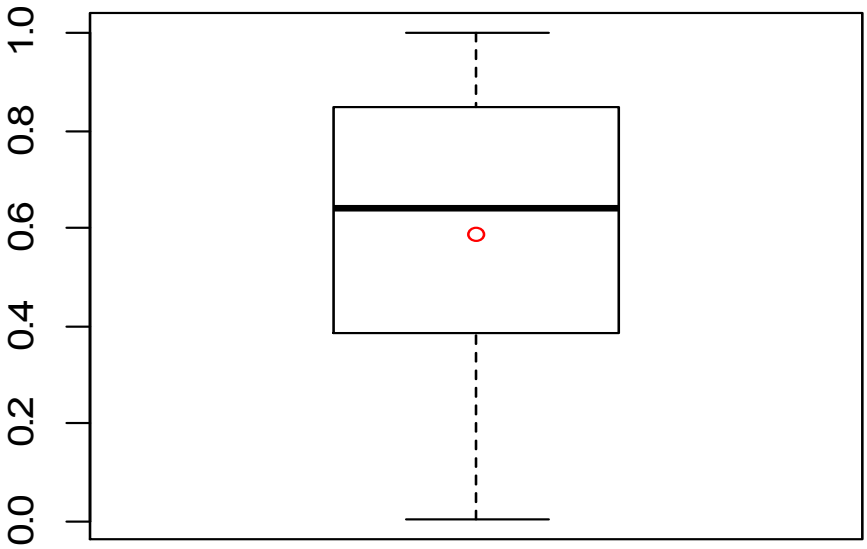

Figure 5.11: Boxplot of the simulated p-values from the variant. The circle indicates the location of the p-value from the observed data.

As we can see from Fig. 5.7 the observed p-value (0.588) is not extreme, and therefore we conclude that the beta-geometric distribution fits the non-smokers data well. We repeat the procedure for the original approach, where the $\boldsymbol{x}_i$ are generated from the $(\hat{\alpha}_i,\hat{\beta}_i)$ which in turn are generated from the asymptotic normal distribution

of the maximum-likelihood estimators from each observed data set.

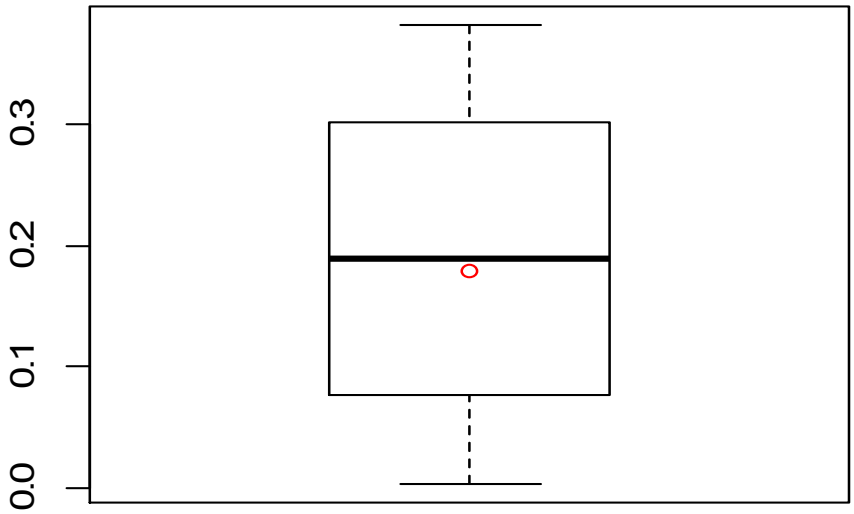

Figure 5.12: Boxplot of the p-values from the original approach. The circle indicates the location of the p-value from the observed data.

As we can see in Fig.5.8 the observed p-value (0.176) is not extreme, and we reach the same conclusion as above, but the sampling distribution of the p-value is now not uniform, so that calibration is required in this case.

CHAPTER 6

METHOD COMPARISON

In this chapter we examine the relative performance of the bootstrap and simulated calibration methods for evaluating the goodness-of-fit of a model. We compare performance by calculating the p-value of each method when the model is true using the same data to find the two p-values. We thus have 500 deviances in the bootstrap method and we calculate how many of these deviances are greater than the observed deviance divided by 500, and we calculate the corresponding p-value using calibrated simulation as described in chapter 5. We repeat the procedure for 100 data sets under the assumed model resulting in 100 pairs of p-values per model, and we generate data under two models: geometric and beta-geometric. The main finding of this chapter is that the p-values from the two methods are very correlated. Fig 6.1 provides a scatter plot of the p-values from the two methods when the assumed model is the geometric distribution. The p-values from the bootstrap are shown on the x-axis and the p-values from the calibrated simulation on the y-axis.

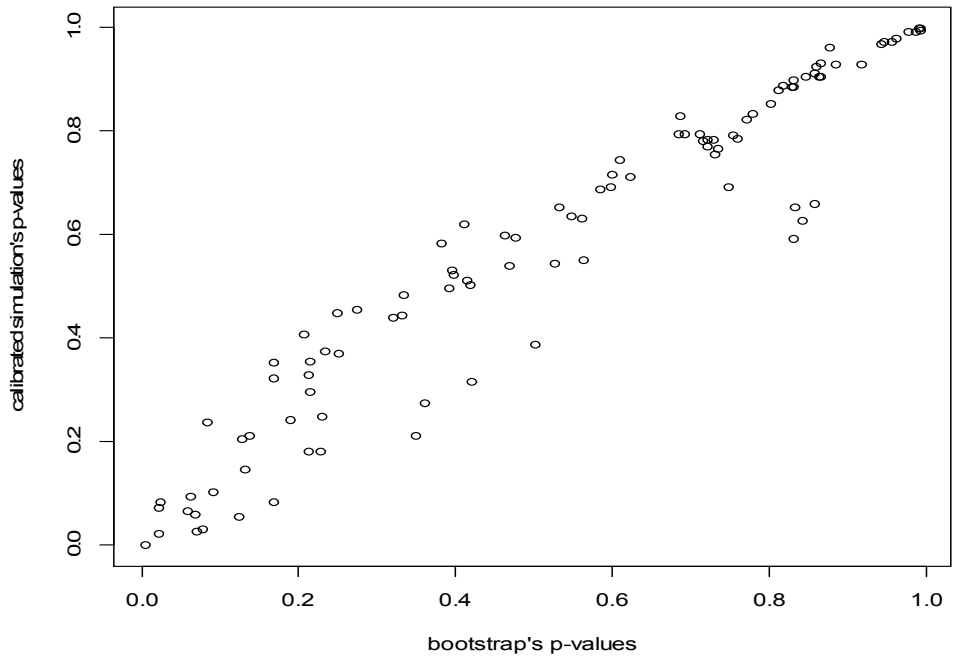

Figure 6.1: Scatter plot of p-values from the bootstrap vs calibrated simulation when the assumed model is the geometric distribution.

As we can see there is a strong positive correlation between the p-values from the two methods Fig. 6.2 provides the corresponding scatter-plot when the assumed model is beta-geometric.

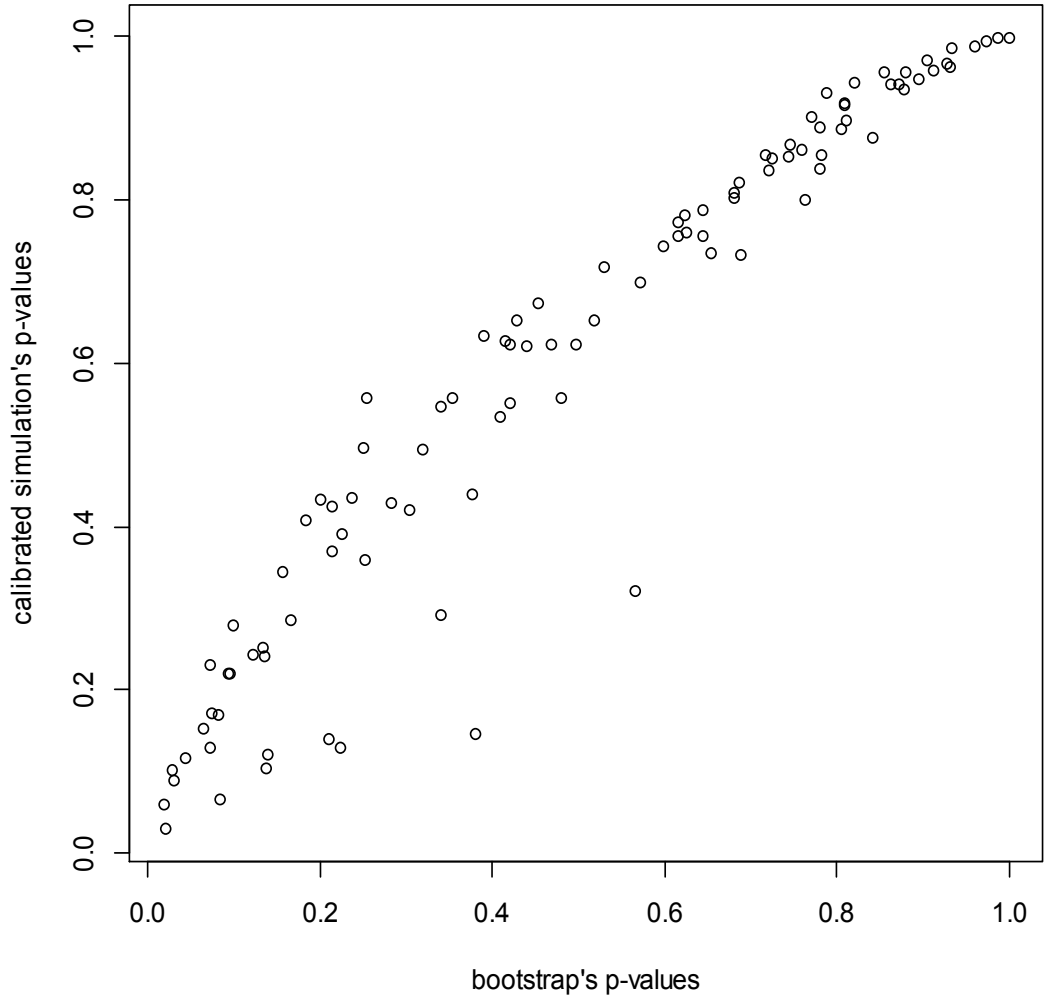

Figure 6.2: Scatter plot of p-values from the bootstrap vs calibrated simulation when the assumed model is the beta-geometric distribution

Again we can see that there is a strong positive correlation between the two methods but there is now an interesting difference in Fig. 6.2.

CHAPTER 7

CAPTURE RECAPTURE MODELS

7.1 Introduction

In a typical capture-recapture experiment in ecology, we place traps or nets in the study area and sample the population several times. At the first trapping sample a number of animals are captured; the animals are uniquely tagged or marked and released into the population. Then at each subsequent trapping sample we record and mark every unmarked animal, record the capture of any animal that has been previously marked, and return all animals to the population. At the end of the experiment the complete capture history for each animal is known. The capture histories of all individuals are arranged in a capture history matrix $\boldsymbol{x}_{\omega}$, as illustrated in Table 7.1:

	Capture occasion j					
Animal i	**1**	**2**	**3**	**4**	**5**	**6**
1	1	1	1	1	1	1
2	1	0	0	1	1	1
3	1	1	0	0	1	1
4	1	1	0	1	1	1
5	1	1	1	1	1	1
6	1	1	0	1	1	1
7	1	1	1	1	1	0
8	1	1	1	0	0	1
9	1	1	1	1	1	1
10	1	1	0	1	1	1

Table 7.1

Such experiments are also called mark-recapture, tag-recapture, and multiple record systems in the literature. The simple type only includes two samples: one is the capture sample and the other the recapture sample.

The capture-recapture technique has been used to estimate population sizes and related parameters such as survival rates, birth rates and migration rates. Biologists

and ecologists recognized that the proportion of previously marked animals in the recapture samples provides a basis for estimating population size. Intuitively, when recaptures in subsequent samples are few, we know that size is much higher than the number of distinct captures. However, if the recapture rate is quite high, then we are likely to have caught most of the animals.

The first use of the capture-recapture technique can be traced back to Laplace, who used it to estimate the population of France in 1786. The earliest applications to ecology include Petersen's work on fish populations in 1896 and Lincoln's work on waterfowl in the 1930's. Capture-recapture has become immensely popular for estimating animal abundance, vital rates and community dynamics for many different species. Currently it is also used in a variety of other research fields including sociology and health science. For example, it is now used to estimate the U.S. Census undercount and the incidence of disease.

The models are generally classified as either closed population or open population models. In a closed population, the size of the population, which is the main interest, is assumed to be constant over the trapping times. The closure assumption is usually valid for data collected in a relatively short time during a non-breeding season. Open populations may have demographic changes (birth or mortality) or migration (immigration or emigration). Open models are usually used to model data from long term studies. Here, in addition to the population size at each sampling time, the parameters of interest also include the survival rates and number of births between sampling times. Here we concentrate on closed models, which also have applications to epidemiology and health science.

7.2 Schnabel census and likelihood functions

The simplest mark-recapture experiment, known as the Lincoln-Petersen procedure, consists of only two samples and provides the most basic estimator for estimating the size *N*.

A natural extension of the Petersen experiment is the so called Schnabel experiment or multiple recapture cencus in which k $(k > 2)$ consecutive samples are taken from the population. If n_i animals are caught in sample i, and m_i are the number found to be marked on a previous sampling occasion, then the u_i $(= n_i - m_i)$ unmarked are

given a mark and the whole sample returned to the population. If individual numbered marks or tags are used, then animals have to be tagged only once, the first time they are caught. Depending whether the n_i are regarded as fixed or random, both the Hyper-geometric and multinomial models readily generalize to this case and for example we have the joint probability:

$$Pr(\{\alpha_\omega\}) = \frac{N!}{\prod_\omega \alpha_\omega!\,(N-r)!} Q^{N-r} \prod_\omega P_\omega^{\alpha_\omega} = \frac{N!}{\prod_\omega \alpha_\omega!\,(N-r)!} \prod_{i=1}^{k} {p_i}^{n_i} {q_i}^{N-n_i}$$

in obvious notation, where α_ω denotes the frequency for observable capture history ω, and r denotes the number of marked animals, N the total population and $p_i (= 1 - q_i)$ is the probability of capture in sample i.

The assumptions underlying the Petersen method must apply to all the samples in a Schnabel cencus so that any departures from these assumptions can seriously affect the validity of $\widehat{N}$. Since variation in catchability seems to be a fact of life, Otis et al (1978) devised a basis of models for estimating N, where these allow capture probabilities to vary with respect to one or more of the factors of time, behavior response, and individual response. In particular, they proposed the following eight models: M_0 (no variation), M_t (variation with time), M_h(variation by individual response or heterogeneity), and various combinations M_{tb}, M_{bh}, M_{th}, and M_{tbh}.

If p_{ij} is the probability that the i-th animal (i=1,2,…,N) is caught in the j-th sample (j=1,2,3,…,k) and we can assume that the animals are independent of one another as far as catching is concerned, then the likelihood function is:

$$\prod_{i=1}^{N} \prod_{j=1}^{k} p_{ij}^{x_{ij}} (1 - p_{ij})^{1-x_{ij}},$$

where x_{ij}=1 if the i-th animal is caught in the j-th sample x_{ij}=0 otherwise. The various models can now be described mathematically by specifying p_{ij}.

7.3 Goodness of fit capture-recapture models

7.3.1 **Model M_0**: $p_{ij} = p$

This is the simplest model where the capture probability is constant over the capture occasions. There are only two parameters in the model, $\theta = (N, p)$, and the joint probability distribution for the data can be written as:

$$Pr(\{x_\omega\}|\theta) = \frac{N!}{\prod_\omega \alpha_\omega!(N-r)!} p^{n.}(1-p)^{kN-n.}$$

where $n. = n_1 + \cdots \ldots + n_k$ denotes the total number of captures. The maximum likelihood estimates of N and p can be obtained using numerical methods. A large – sample variance for $\widehat{N}$ is given by (Darroch 1958)

$$Var(\widehat{N}) = \frac{N}{(1-p)^{-k} + (k-1) - k(1-p)^{-1}}$$

White et al (1982, p48) provide a complete capture history matrix from model M_0 with $N = 50, p = 0.3$ and $k = 6$ occasions:

	Occasions					
	1	2	3	4	5	6
1	1	1	1	1	0	0
2	1	0	0	0	0	0
3	1	0	1	0	0	1
4	1	0	0	0	0	1
5	1	0	0	0	0	0
6	1	1	0	0	0	0
7	1	1	0	0	0	0
8	1	0	1	0	1	1
9	1	0	0	0	1	0
10	1	1	1	0	0	0
11	1	0	0	0	0	0
12	1	0	0	0	0	0
13	1	0	0	1	0	0
14	1	0	0	1	1	0
15	1	0	1	0	0	0
16	1	0	1	0	0	0
17	0	1	0	0	0	1
18	0	1	0	0	0	1
19	0	1	0	0	1	0
20	0	1	0	0	0	0
21	0	1	1	1	0	1
22	0	1	0	0	1	1
23	0	1	0	0	1	0
24	0	0	1	0	1	0
25	0	0	1	0	0	0

26	0	0	1	0	0	1
27	0	0	1	0	0	0
28	0	0	1	1	0	0
29	0	0	1	0	1	0
30	0	0	1	0	0	1
31	0	0	1	0	0	1
32	0	0	0	1	0	0
33	0	0	0	1	0	0
34	0	0	0	1	0	0
35	0	0	0	1	0	1
36	0	0	0	1	0	0
37	0	0	0	1	0	1
38	0	0	0	1	1	0
39	0	0	0	1	1	1
40	0	0	0	1	0	0
41	0	0	0	0	1	0
42	0	0	0	0	1	0
43	0	0	0	0	1	1
44	0	0	0	0	1	1
45	0	0	0	0	0	1
46	0	0	0	0	0	1
47	0	0	0	0	0	1

table 7.2: Capture-recapture data from White et al (1982).

We are going to check the goodness of fit of the M_0 model. The capture-recapture summary statistics from the 6 occasions are: n_1=16, n_2=11, n_3=15, n_4=14, n_5=14, n_6=18, $r = 47$ for each occasion, then we are taking the likelihood from above and we are finding the log – likelihood which is:

$$\log(L) = \log \mathrm{N}! - \log(N - r) + (\sum_{i=1}^{6} n_i)\log(p) + (6N - (\sum_{i=1}^{6} n_i))\log(1 - p)$$

The maximum likelihood estimates of **θ** are:

Estimate of N	Estimate of p
55.247	**0.265**

And the hessian matrix is:

$$\mathrm{H}(\hat{\theta}) = \begin{bmatrix} 0.09625797 & 8.168589 \\ 8.168589 & 1699.935180 \end{bmatrix}$$

resulting in the variance-covariance matrix of $\hat{\theta}$:

$$Var(\hat{\theta}) = \begin{bmatrix} 17.542 & -0.084 \\ -0.084 & 0.001 \end{bmatrix}$$

We proceed by generating parameters values $\widehat{N}_i$, $\hat{p}_i$ from the asymptotic normal distribution of the maximum-likelihood estimates $\widehat{N}$, $\hat{p}$. Thus

$$\begin{bmatrix} \widehat{N}_i \\ \hat{p}_i \end{bmatrix} \sim \mathrm{N}\left(\begin{bmatrix} 55.247 \\ 0.265 \end{bmatrix}, \begin{bmatrix} 17.542 & -0.084 \\ -0.084 & 0.001 \end{bmatrix}\right), i=1,2,\ldots,500$$

We are going to demonstrate a few of the simulated parameter values:

i	$\widehat{N}_i$	$\hat{p}_i$
1	57.012	0.243
2	58.536	0.213
3	50.736	0.288
4	54.713	0.226
5	52.673	0.299
6	51.52	0.318
7	54.842	0.289
8	60.321	0.191
9	53.086	0.275
10	57.826	0.243
11	55.396	0.262
12	50.277	0.3
13	51.651	0.314
14	64.335	0.247
15	56.971	0.243
16	55.457	0.238
17	59.188	0.204
18	58.444	0.231
19	60.051	0.245
20	46.318	0.303

For each pair of simulated parameters values ($\widehat{N}_i$, $\hat{p}_i$) we simulate data sets $\boldsymbol{x}_i$ from the model, of dimensions matched to the observed data.

The code in the R programming language is interesting because I have to generate 500 matrices and then calculate the capture-recapture summary statistics n_i and r of each matrix. Let's see an example of the code:

```
M<-matrix(rbinom(round(Nhat)*6,1,phat),nrow=round(Nhat),ncol=6)
indx<-apply(M,1,sum)>0
m<-M[indx,]
m0<-apply(m,2,sum)
r0<-nrow(m)
```

where the variables `Nhat` and `phat` contain the parameter values of N and p respectively.

The first 7 out of a total of 500 simulated data sets are shown below:

Number of animals captured the day j, j=1,…6	Simulated sets x_i using $(\widehat{N}_i, \widehat{p}_i)$, i=1,2,…,7						
	1	**2**	**3**	**4**	**5**	**6**	**7**
n_1	13	15	14	11	16	13	19
n_2	11	15	16	12	15	14	14
n_3	15	17	13	8	14	17	13
n_4	8	11	12	12	15	17	21
n_5	14	16	17	13	15	13	12
n_6	13	11	15	20	12	22	13

There are several ways to form expected values for a capture-recapture study. On the average for model M_0, we would expect to catch $E[n_j] = Np$ animals on the j^{th} occasion, and we can readily compute expected number of animals caught at each occasion for the observed and simulated data sets.

The expected values for the 7 simulated data sets above are:

Expected frequencies e_i, i=1,…,7						
13.832	12.476	14.595	12.392	15.758	16.376	15.842

We then calculate a measure of the discrepancy between the observed data $\boldsymbol{x}$ and the assumed model using the Freeman-Tukey measure. Here we demonstrate the first 125 values:

$D(x;\widehat{N}_i,\widehat{p}_i)$, i=1,2,…,125				
0.54	0.481	0.789	0.555	0.518
0.974	0.697	0.797	0.479	0.522
0.479	0.555	0.481	0.513	0.515
1.016	0.771	0.48	0.832	0.483
0.615	0.525	0.573	0.489	1.146
0.79	0.5	1.177	0.592	0.52
0.635	0.885	0.505	0.814	0.694
1.544	1.154	0.514	0.914	0.764
0.479	0.66	0.613	0.518	0.669
0.512	0.722	0.684	1.007	0.525
0.48	0.585	0.626	0.596	0.715
0.505	0.482	0.999	0.68	0.571
0.739	0.48	0.803	0.484	0.519
0.644	0.571	0.712	0.785	0.57
0.533	0.681	0.531	0.601	0.539
0.689	0.551	0.672	0.631	0.753
1.201	1.505	1.044	0.48	0.736
0.607	0.637	0.655	1.515	0.594
0.481	0.48	0.719	0.49	0.498
0.51	0.997	0.482	0.539	0.645
0.587	0.553	0.772	0.54	0.657
0.48	0.487	0.483	1.143	0.491
0.48	0.51	0.589	0.486	0.551
0.977	0.837	0.627	0.519	0.62
0.567	0.77	0.574	0.915	0.494

We also calculate the Freeman–Tukey discrepancy measure between the simulated data set $\boldsymbol{x}_i$ and the assumed model $\mathrm{D}(x_i;\widehat{N}_i,\hat{p}_i)$. The first 125 discrepancy values are shown below:

$D(x_i;\widehat{N}_i,\widehat{p}_i)$ i=1,2,...,125				
1.005	0.285	0.236	0.15	0.85
0.893	2.386	1.294	1.062	0.49
0.306	0.879	0.468	1.001	1.849
1.44	1.516	0.786	1.353	1.612
0.337	0.176	0.618	2.637	1.289
0.908	1.028	0.913	1.238	1.196
1.11	1.231	0.485	1.823	0.658
0.536	0.436	1.999	2.285	0.753
2.601	2.332	0.993	0.219	1.6
1.633	0.502	0.42	1.701	0.373

1.835	0.567	0.552	0.927	0.669
0.31	2.013	1.279	0.737	1.585
0.755	1.524	0.378	1.119	0.357
0.308	1.038	0.75	2.209	1.122
1.942	2.125	1.236	1.055	0.585
1.229	2.241	0.651	0.738	0.195
0.334	1.834	0.365	0.367	2.086
1.86	0.709	0.809	0.916	1.778
2.308	1.943	0.438	1.034	0.803
1.069	0.943	1.722	0.613	1.539
0.293	2.191	1.557	0.389	1.441
0.652	0.515	3.222	1.34	0.86
1.205	0.456	0.659	0.966	0.48
1.887	0.466	1.501	0.616	0.602
0.903	0.414	1.561	1.312	2.812

As in Chapter 6, the observed and simulated discrepancy values $\mathrm{D}(x; \hat{N}_i, \hat{p}_i)$ and $\mathrm{D}(x_i; \hat{N}_i, \hat{p}_i)$ are in matched pairs e.g. 0.54 is matched to 1.005. The p-value of the method for the goodness of fit of model M_0 to the data in table 7.1 is thus 0.736 resulting in the following plot:

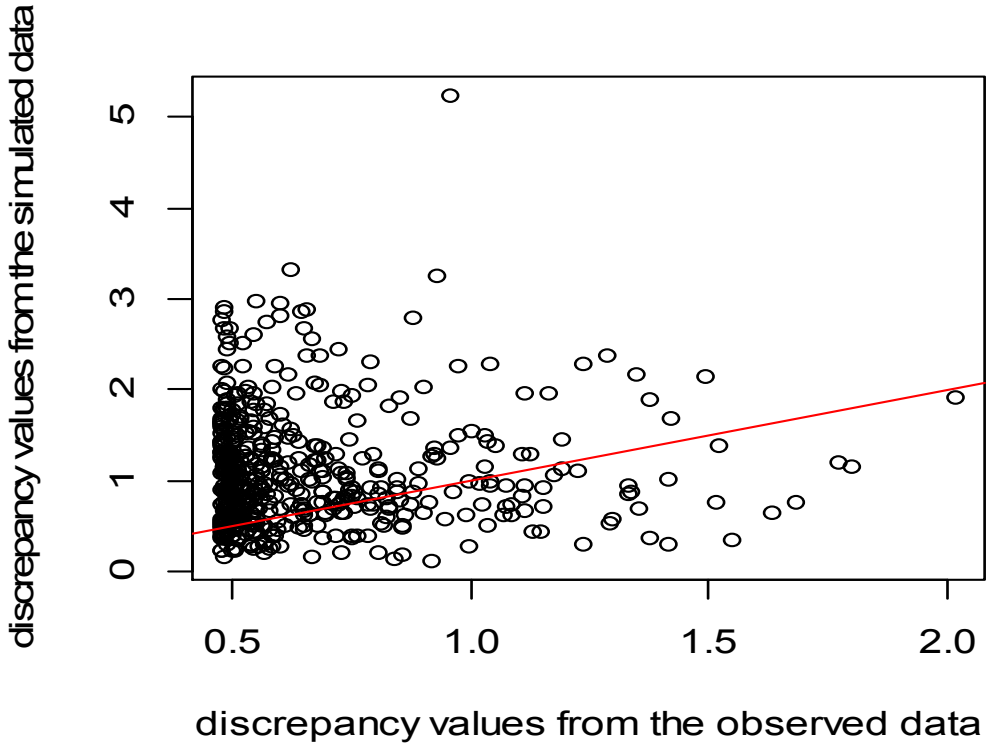

Figure 7.1: Scatter plot of $\mathrm{D}(x; \hat{N}_i, \hat{p}_i)$ vs $\mathrm{D}(x_i; \hat{N}_i, \hat{p}_i)$

We now consider a new variant of the method where the simulated data sets $\boldsymbol{x}_i$, are obtained using the maximum likelihood point estimates $\hat{N}$, $\hat{p}$, as opposed to random values from the asymptotic normal distribution.

We simulate 500 data sets, as above. The first 7 of these data sets are presented below:

Number of animals captured the day j, j=1,…6	Simulated sets x_i using $\hat{N}, \hat{p}$, i =1,…,7						
	1	2	3	4	5	6	7
n_1	15	23	19	19	13	15	21
n_2	13	13	19	13	20	13	13
n_3	13	15	16	12	9	9	13
n_4	15	14	16	14	18	9	14
n_5	10	11	15	14	11	11	11
n_6	16	16	22	11	11	17	11

The expected number of animals caught at occasion j is 14.667 for all 500 sets because the expected value is $E(n_j) = \hat{N} \cdot \hat{p}$ and in this case in particular 55.247· 0.265 which is the same for each simulated data set $\boldsymbol{x}_i$ as a result of all depending on $\hat{N}$,$\hat{p}$. The Freeman–Tukey discrepancy measure between the observed data and the model is $D(\boldsymbol{x};\hat{N},\hat{p}\ ) =$ 0.4802. We are going to use that value 500 times as we shall see below.

As above we also calculate the Freeman–Tukey discrepancy measure between the simulated data set $\boldsymbol{x}_i$ and the model $D(\boldsymbol{x}_i;\hat{N},\hat{p})$ and we provide example values below for i=1,…,150 of 500 which we have actually calculated:

$D(\boldsymbol{x}_i;\hat{N},\hat{p})$ i=1,…,150				
0.579	0.834	1.356	1.621	0.829
1.285	0.252	2.036	1.763	1.154
1.361	0.798	1.135	1.088	0.344
0.743	0.533	1.853	0.612	1.82
1.848	0.975	0.847	0.344	0.186
1.778	4.588	0.606	0.402	1.318
1.202	1.776	1.17	0.852	0.381
0.685	0.225	1.363	0.919	0.836
0.972	0.518	2.822	0.807	0.23
0.593	2.1	1.866	0.719	0.715

0.702	0.935	1.17	2.22	1.003
0.408	1.388	1.015	1.719	0.408
2.491	0.665	1.297	1.591	0.711
0.627	1.262	0.914	2.321	1.145
1.296	0.125	0.701	1.334	0.454
0.498	0.774	1.645	1.835	1.198
1.093	2.544	2.123	0.548	1.454
1.818	1.074	0.915	0.678	0.523
0.968	2.092	1.214	0.372	1.205
0.392	1.999	0.974	0.776	1.537
0.218	0.747	0.555	0.393	1.169
1.637	0.203	0.93	0.471	0.823
0.513	2.462	2.407	0.521	0.585
0.943	0.809	0.702	1.198	1.282
0.792	0.956	0.588	0.514	1.663
1.132	2.576	0.355	1.34	1.322
1.616	2.416	0.436	2.014	1.927
1.387	0.959	0.657	2.468	2.866
0.508	1.192	0.309	0.57	1.161

Fig. 7.2 provides a scatter plot of $D(\boldsymbol{x}; \widehat{N}, \hat{p})$ vs $D(\boldsymbol{x}_i; \widehat{N}, \hat{p})$. It is obvious that the points lie on a straight line since we use the maximum likelihood estimates. The p-value of the variant is 0.868 which is in agreement with the original approach.

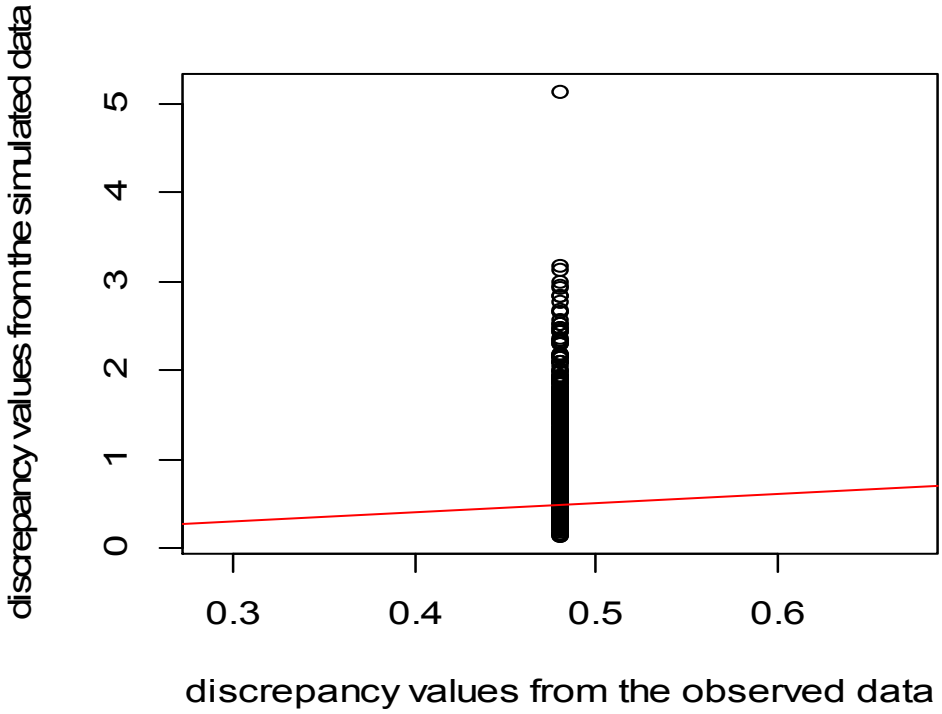

Figure 7.2: Scatter plot of D(x;$\widehat{N}\hat{p}$) vs D(x_i;$\widehat{N}\hat{p}$) from the variant

We calibrate the observed p-values, as described in Sections 5.5.1.2 and 5.5.2.2. Thus we are going to generate 100 simulated "observed" data sets from which we are going to find 100 p-values for each case. The 100 simulated "observed" data sets are going to be generated using the M.L.E. from the observed data and the binomial distribution and we are going to calibrate the p-value from the original approach as well the new variant. First let's calibrate the p-value from the original approach where the data sets x_i's are generated from $\widehat{N}_i, \hat{p}_i$ which in turn are generated from the asymptotic normal distribution of the maximum likelihood estimators from each "observed" data set. Fig 7.3 provides a boxplot of the simulated p-values.

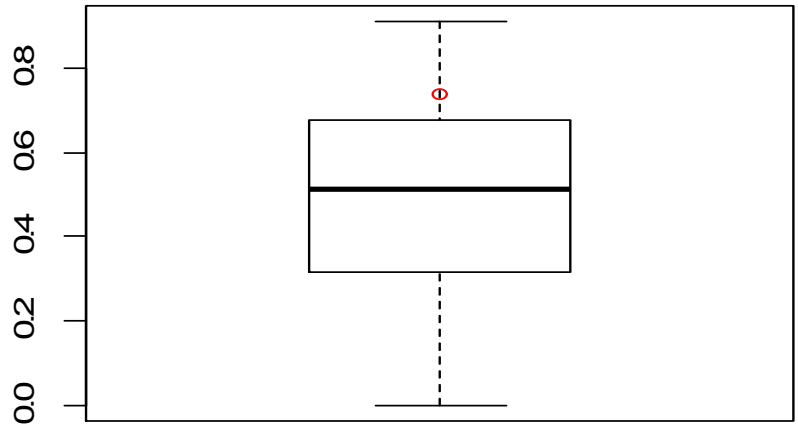

Figure 7.3: Boxplot of the simulated p-values. The circle indicates the location of the p-value from the observed data.

We can see the observed p-value (0.736) is above the box so we could say that the model is fitted well and the M_0 is a good model for our data. The corresponding boxplot from the variant where the x_i's are generated from the M.L.E. is presented in Fig7.4:

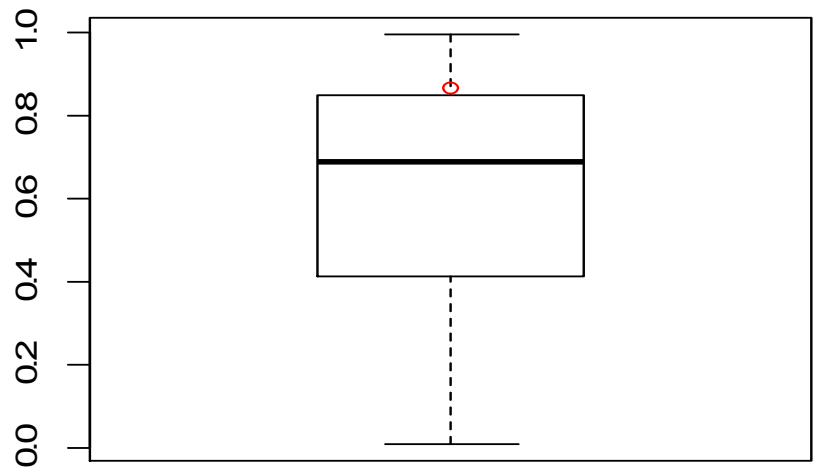

Figure 7.4: Boxplot of the simulated p-values from the variant. The circle indicates the location of the p-value from the observed data.

We can see that the observed p-value (0.868) is above the black line in the boxplot so we could say that our model is good and fits well our data.

7.3.2 **Model M_t:** $p_{ij} = p_j$

This model allows the capture probabilities to vary with time and reduces to the Lincoln-Petersen model when $k = 2$. In general, there are $k + 1$ parameters, $(N, p_1, \dots, p_k)$, and the (multinomial) likelihood is given from:

$$Pr(\{a_\omega\}) = \frac{N!}{\prod_\omega a_\omega!(N-r)!} Q^{N-r} \prod_\omega P_\omega^{a_\omega} = \frac{N!}{\prod_\omega a_\omega!(N-r)!} \prod_{j=1}^{k} {p_j}^{n_j} {q_j}^{N-n_j}$$

The log – likelihood which derives from above is:

$$\log L = \log N! - \log(N-r)! + \sum_{j=1}^{5} n_j \log \frac{n_j}{N} + \sum_{j=1}^{5} (N - n_j) \log\left(1 - \frac{n_j}{N}\right) \quad \text{(eq.7.1)}$$

We need to mention that in examining the goodness of fit of this model we have to do some small changes in comparison to M_0. In Section 7.3.1 we used the observed and

expected number of captures at each occasion in order to examine the goodness of fit of the model. In the M_t model this is not possible because the expected value at occasion j is $\widehat{N}\hat{p}_j$, which coincides with the observed number n_j. So in this model we are going to use the capture frequencies f_j instead where f_j represents the number of animals captured j times. We shall have the observed frequencies and the expected frequencies and we shall continue doing our work the same way as in previous sections. First we need some capture – recapture data which are the following based on capture-recapture summary statistics provided by White et al (1982, p52):

Animal	Occasion j				
	1	2	3	4	5
1	1	1	1	1	0
2	1	1	1	1	0
3	1	1	1	1	0
4	1	1	1	0	0
5	1	1	1	0	0
6	1	1	1	0	0
7	1	1	1	0	0
8	1	1	1	0	0
9	1	1	1	0	0
10	1	1	1	0	0
11	1	1	1	0	0
12	1	1	1	0	0
13	1	1	1	0	0
14	1	1	1	0	0
15	1	1	1	0	0
16	1	1	1	0	0
17	1	1	1	0	0
18	1	1	1	0	0
19	1	1	1	0	0
20	1	1	1	0	0
21	1	1	1	0	0
22	1	1	1	0	0
23	1	1	1	0	0
24	1	1	0	0	0
25	1	1	0	0	0
26	1	1	0	0	0
27	1	1	0	0	0
28	1	1	0	0	0
29	1	1	0	0	0
30	1	1	0	0	0
31	1	1	0	0	0

32	1	1	0	0	0
33	1	1	0	0	0
34	1	1	0	0	0
35	1	1	0	0	0
36	1	1	0	0	0
37	1	1	0	0	0
38	1	1	0	0	0
39	1	1	0	0	0
40	1	1	0	0	0
41	1	1	0	0	0
42	1	1	0	0	0
43	1	1	0	0	0
44	1	1	0	0	0
45	1	1	0	0	0
46	1	1	0	0	0
47	1	1	0	0	0
48	1	1	0	0	0
49	1	1	0	0	0
50	1	1	0	0	0
51	1	1	0	0	0
52	1	1	0	0	0
53	1	1	0	0	0
54	1	1	0	0	0
55	1	1	0	0	0
56	1	1	0	0	0
57	1	1	0	0	0
58	1	1	0	0	0
59	1	1	0	0	0
60	1	1	0	0	0
61	1	1	0	0	0
62	1	1	0	0	0
63	1	1	0	0	0
64	1	1	0	0	0
65	1	1	0	0	0
66	1	1	0	0	0
67	1	1	0	0	0
68	1	1	0	0	0
69	1	1	0	0	0
70	1	1	0	0	0
71	1	0	1	0	0
72	1	0	1	0	0
73	1	0	1	0	0
74	1	0	1	0	0
75	1	0	1	0	0

76	0	0	1	0	0
77	0	0	1	0	0
78	0	0	1	0	0
79	0	0	1	0	0
80	0	0	1	0	0
81	0	0	1	0	0
82	0	0	1	0	0
83	0	0	1	0	0
84	0	0	1	0	0
85	0	0	1	0	0
86	0	0	1	0	0
87	0	0	1	0	0
88	0	0	1	0	0
89	0	0	1	0	0
90	0	0	1	0	0
91	0	0	1	0	0
92	0	0	1	0	0
93	0	0	1	0	0
94	0	0	1	0	0
95	0	0	1	0	0
96	0	0	1	0	0
97	0	0	1	0	0
98	0	0	1	0	0
99	0	0	1	0	0
100	0	0	1	0	0
101	0	0	0	1	0
102	0	0	0	1	0
103	0	0	0	1	0
104	0	0	0	1	0
105	0	0	0	1	0
106	0	0	0	1	0
107	0	0	0	1	0
108	0	0	0	1	0
109	0	0	0	1	0
110	0	0	0	1	0
111	0	0	0	1	0
112	0	0	0	1	0
113	0	0	0	1	0
114	0	0	0	1	0
115	0	0	0	1	0
116	0	0	0	1	0
117	0	0	0	1	0
118	0	0	0	1	0
119	0	0	0	1	0

120	0	0	0	1	0
121	0	0	0	1	0
122	0	0	0	0	1
123	0	0	0	0	1
124	0	0	0	0	1
125	0	0	0	0	1
126	0	0	0	0	1
127	0	0	0	0	1

Table 7.3

The summary statistics here are $n_1 = 29$, $n_2 = 56$, $n_3 = 61$, $n_4 = 52$, $n_5 = 37$. These are the numbers of animal caught in 5 different occasions and the total number of animals caught respectively. The maximum likelihood estimate of N is from Eq.7.1:

Estimate of *N*	**Value of log - likelihood**
132.7933	**-103.5676**

The estimated probabilities for each occasion are given by $\hat{p}_j = \frac{n_j}{\hat{N}}$, j=1,2,3,4,5, the results are $\hat{p}_1$= 0.186, $\hat{p}_2$= 0.352, $\hat{p}_3$= 0.332, $\hat{p}_4$= 0.398, $\hat{p}_5$= 0.245

and the variance-covariance matrix of $\hat{\theta} = (\hat{N}, \hat{p}_1, \ldots, \hat{p}_5)$ is:

$$Var(\hat{\theta}) = \begin{pmatrix} 0.036 & 1.228 & 1.542 & 1.496 & 1.661 & 1.325 \\ 1.228 & 996.551 & 0 & 0 & 0 & 0 \\ 1.542 & 0 & 661.154 & 0 & 0 & 0 \\ 1.496 & 0 & 0 & 679.952 & 0 & 0 \\ 1.661 & 0 & 0 & 0 & 629.077 & 0 \\ 1.325 & 0 & 0 & 0 & 0 & 813.825 \end{pmatrix}$$

To check the goodness of fit of the model, we proceed by generating parameter values θ_i from the asymptotic normal distribution of $\hat{\theta}$

$$\hat{\theta}_i \sim \mathrm{N}\left(\hat{\theta}, Var(\hat{\theta})\right), i = 1, \ldots, 500$$

and for each simulated set of parameter values we simulate data sets $\boldsymbol{x}_i$ from the model.

The first 13 simulated parameter values $\hat{\theta}_i$ are shown below:

i	$\hat{N}_i$	$\hat{p}_{1i}$	$\hat{p}_{2i}$	$\hat{p}_{3i}$	$\hat{p}_{4i}$	$\hat{p}_{5i}$
1	148.672	0.201	0.356	0.317	0.427	0.235
2	156.678	0.211	0.406	0.325	0.374	0.302
3	150.201	0.125	0.385	0.225	0.37	0.284
4	148.177	0.2	0.315	0.39	0.402	0.201
5	150.536	0.168	0.356	0.362	0.481	0.209
6	151.321	0.201	0.283	0.346	0.396	0.253
7	142.422	0.209	0.352	0.304	0.354	0.21
8	152.605	0.156	0.32	0.283	0.305	0.257
9	144.964	0.228	0.32	0.366	0.386	0.246
10	158.116	0.184	0.377	0.37	0.359	0.227
11	157.153	0.137	0.351	0.289	0.394	0.23
12	154.017	0.183	0.374	0.289	0.423	0.283
13	146.933	0.127	0.378	0.339	0.366	0.209

We next calculate the capture frequencies f_j representing the number of animals captured j times. The probability of an animal to be captured the j times in 5 occasions is:

$$Pr(caught\ 1\ time) = p_1(1-p_2)\ (1-p_3)(1-p_4)(1-p_5) + (1-p_1)\ p_2(1-p_3)\ (1-p_4)(1-p_5) + (1-p_1)(1-p_2)p_3(1-p_4)(1-p_5) + (1-p_1)(1-p_2)\ (1-p_3)p_4(1-p_5) + (1-p_1)(1-p_2)(1-p_3)(1-p_4)p_5.$$

$$Pr(caught\ 2\ times) = p_1p_2(1-p_3)(1-p_4)(1-p_5) + p_1(1-p_2)p_3(1-p_4)(1-p_5) + p_1(1-p_2)(1-p_3)p_4(1-p_5) + p_1(1-p_2)(1-p_3)(1-p_4)p_5 + (1-p_1)p_2p_3(1-p_4)(1-p_5) + (1-p_1)p_2(1-p_3)p_4(1-p_5) + (1-p_1)p_2(1-p_3)(1-p_4)p_5 + (1-p_1)(1-p_2)p_3p_4(1-p_5) + (1-p_1)(1-p_2)p_3(1-p_4)p_5 + (1-p_1)(1-p_2)(1-p_3)p_4p_5.$$

p41−p5+p1(1−p2)

$$Pr(caught\ 3\ times) = p_1p_2p_3(1-p_4)(1-p_5) + p_1p_2(1-p_3)\ p_4(1-p_5) + p_1p_2(1-p_3)(1-p_4)p_5 + p_1(1-p_2)p_3p_4(1-p_5) + p_1(1-p_2)\ (1-p_3)\ p_4p_5 + (1-p_1)p_2(1-p_3)p_4p_5 + (1-p_1)p_2p_3p_4(1-p_5) + (1-p_1)(1-p_2)p_3p_4p_5.$$

$$Pr(caught\ 4\ times) = p_1p_2p_3p_4(1-p_5) + (1-p_1)p_2p_3p_4p_5 + p_1(1-p_2)p_3p_4p_5 + p_1p_2(1-p_3)p_4p_5 + p_1p_2p_3(1-p_4)p_5.$$

$$Pr(caught\ 5\ times) = p_1p_2p_3p_4p_5$$

So the discrete probability distribution of a sum of independent Bernoulli trials that are not necessarily identically distributed is the Poisson binomial distribution. Now we can calculate the expected frequencies since we know that the number of animals caught in 5 trials follows the Poisson binomial distribution. Similarly we calculate the observed frequencies f_j from the observed data $\boldsymbol{x}$**:**

Frequencies f_j from the observed data				
1	**2**	**3**	**4**	**5**
55	**48**	**20**	**4**	**0**

and the 500 simulated data sets x_i. The observed frequencies from the first 7 simulated data sets are shown below:

f_{ji} 's from the simulated data sets x_i using the $\hat{N}_i, \hat{p}_{ji}$, i=1,…,500, j=1,…,5							
	1	**2**	**3**	**4**	**5**	**6**	**7**
f_{1i}	51	59	47	43	53	52	44
f_{2i}	43	47	46	51	51	46	48
f_{3i}	18	18	21	28	22	19	24
f_{4i}	4	3	4	7	4	4	6
f_{5i}	0	0	0	1	0	0	0

The corresponding expected frequency values for these 7 data sets are:

Expected values of frequencies (e_{ji}) i=1,…,500, j=1,…,5							
	1	**2**	**3**	**4**	**5**	**6**	**7**
e_{1i}	60.167	62.3	57.782	61.494	63.923	60.107	57.44
e_{2i}	38.39	45.229	42.857	46.795	42.035	42.192	45.812
e_{3i}	10.318	14.04	13.606	14.586	11.293	12.696	14.8
e_{4i}	0.949	1.568	1.53	1.508	1.003	1.362	1.505
e_{5i}	0	0	0	0	0	0	0

For each set of expected frequencies, we then calculate a measure of the discrepancy between the observed frequencies f_j and the assumed model using the Freeman-Tukey measure.

$D\left(f_j; \hat{N}_i, \hat{p}_{ji}\right)$, j=1,…5, i=1,…,65				
3.292	4.834	2.554	1.745	3.392

1.353	0.424	0.886	2.851	2.302
1.376	1.207	1.064	0.53	1.346
1.211	3.233	1.633	0.871	0.939
2.766	1.41	1.63	2.655	2.341
1.821	2.565	1.552	3	1.692
1.04	0.994	1.471	1.717	3.461
1.442	1.346	1.975	0.581	4.635
1.518	1.447	2.493	3.661	2.586
2.125	2.935	1.479	1.377	1.674
1.814	3.788	1.678	2.705	0.406
1.136	2.388	2.243	0.236	1.932
1.529	1.512	0.418	3.284	2.761

and we also calculate the Freeman–Tukey discrepancy measure $D(f_{ji};\widehat{N}_i,\hat{p}_{ji})$ between the simulated frequencies f_{ji} and the assumed model. The first 65 discrepancy values are shown in each case.

$D(f_{ji};\widehat{N}_i,\hat{p}_{ji})$ j=1,…5, i=1,…,65				
2.624	1.203	0.312	1.46	3.049
0.538	5.306	5.369	3.312	2.287
1.993	4.735	2.235	5.435	1.477
6.918	1.377	3.567	1.921	1.74
3.71	2.47	3.738	1.474	6.384
1.703	1.49	1.591	3	5.586
3.521	1.19	0.371	4.871	1.89
5.67	2.408	1.093	0.171	1.529
4.966	1.874	1.245	0.705	1.055
3.065	2.743	1.53	0.412	1.359
1.773	1.728	2.747	2.623	3.766
0.371	6.491	2.39	0.277	4.82
0.436	0.613	1.402	1.991	1.388
1.16	2.943	2.004	2.273	0.984
2.291	1.108	3.282	3.363	4.494
3.49	1.952	1.186	1.735	1.341
5.588	1.2	0.297	0.538	4.961
1.053	0.49	0.812	6.461	1.009
2.959	6.041	0.971	2.663	3.643
4.73	1.83	2.656	1.522	6.919

Fig. 7.5 provides a scatter plot of $D(f_j;\widehat{N}_j,\hat{p}_{ji})$ vs $D(f_{ji};\widehat{N}_i,\hat{p}_{ji})$. The p-value of the method is 0.634, and as such model M_t appears to fit the data well.

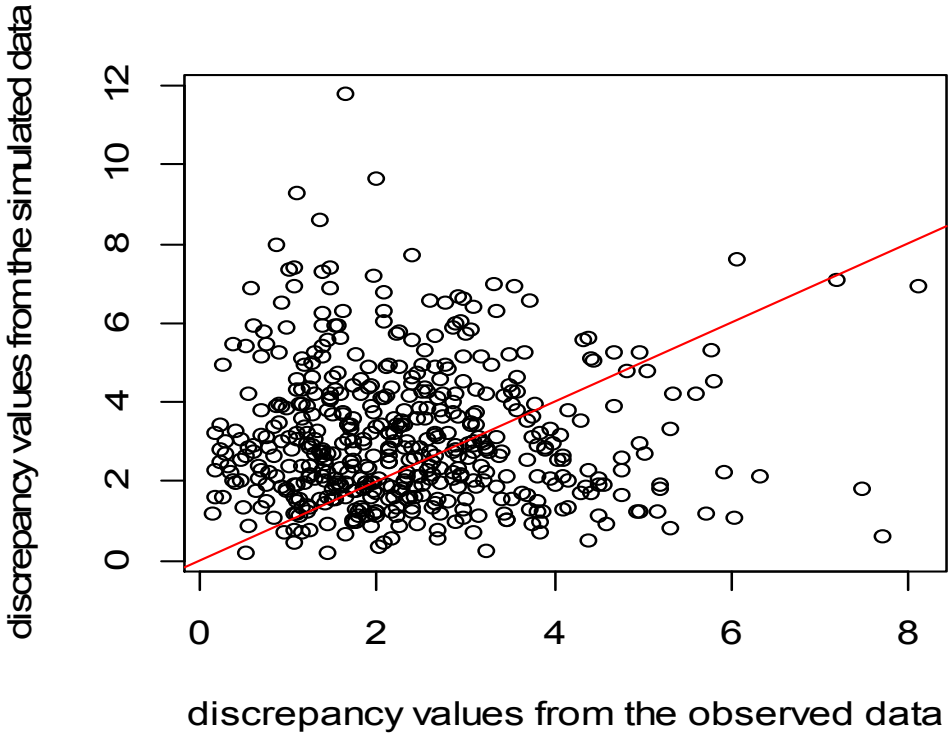

Figure 7.5: Scatter plot of values $D(f_j; \hat{N}_j, \hat{p}_{ji})$ vs $D(f_{ji}; \hat{N}_i, \hat{p}_{\mathrm{ji}})$

We now consider the variant of the method where the observed and simulated discrepancy values are based on the maximum likelihood point estimates. We thereby simulate 500 data sets $\boldsymbol{x_i}$ using $\hat{\theta}$ and calculate simulated frequencies f_{ji}. The first 7 sets of simulated frequencies are shown below:

f_{ji} 's from the simulated data sets x_i using the $\hat{N}$ and $\hat{p}_j$, j=1,2,3,4,5, i=1,2,…,7							
	1	**2**	**3**	**4**	**5**	**6**	**7**
f_1	52	54	58	54	56	54	59
f_2	47	50	47	44	46	48	51
f_3	20	21	19	18	18	21	22
f_4	4	4	4	3	4	4	4
f_5	0	0	0	0	0	0	0

We next calculate the expected frequencies from the original observed data:

Expected values of frequencies (e_j) j=1,2,3,4,5				
54.428	47.61	20.127	4.093	0.319

which are also the same for each simulated data set f_{ji} as a result of all depending on $\widehat{N}$ and $\hat{p}_j, j$=1,2,3,4,5. The Freeman–Tukey discrepancy measure between the observed frequencies f_j and the model $D(f_j; \widehat{N}_j, \hat{p}_{ji})$ = 0.529 and we are going to use that value 500 times as we shall see below.

We also calculate the Freeman–Tukey discrepancy measure between the simulated frequencies f_{ji} and the model $D(f_{ji}; \widehat{N}_i, \hat{p}_{\text{ji}})$and we provide the first 70 of 500 discrepancy values below.

$D(f_{ji}; \widehat{N}_i, \hat{p}_{\text{ji}})$, j=1,…,5, i=1,…,70				
0.35	0.545	1.115	0.385	0.402
0.359	0.444	0.331	0.389	0.359
0.395	0.807	0.431	0.479	0.612
0.535	0.498	0.584	0.384	0.348
0.404	0.658	0.584	1.035	0.387
0.331	0.665	0.402	0.85	0.68
0.512	0.606	0.331	0.36	0.677
0.796	0.348	1.623	0.435	0.774
0.351	0.332	0.847	0.395	1.12
0.606	0.463	0.417	0.429	0.66
0.463	0.4	0.604	0.366	0.349
0.463	0.441	0.347	1.13	0.545
0.964	0.978	0.415	0.361	0.507
0.99	1.228	0.522	1.093	0.382

Fig. 7.6 provides a scatter plot of $D(f_j; \widehat{N}_j, \hat{p}_{ji})$ vs $D(f_{ji}; \widehat{N}_i, \hat{p}_{\text{ji}})$. The p-value of the variant is 0.872 which is in agreement with the original approach.

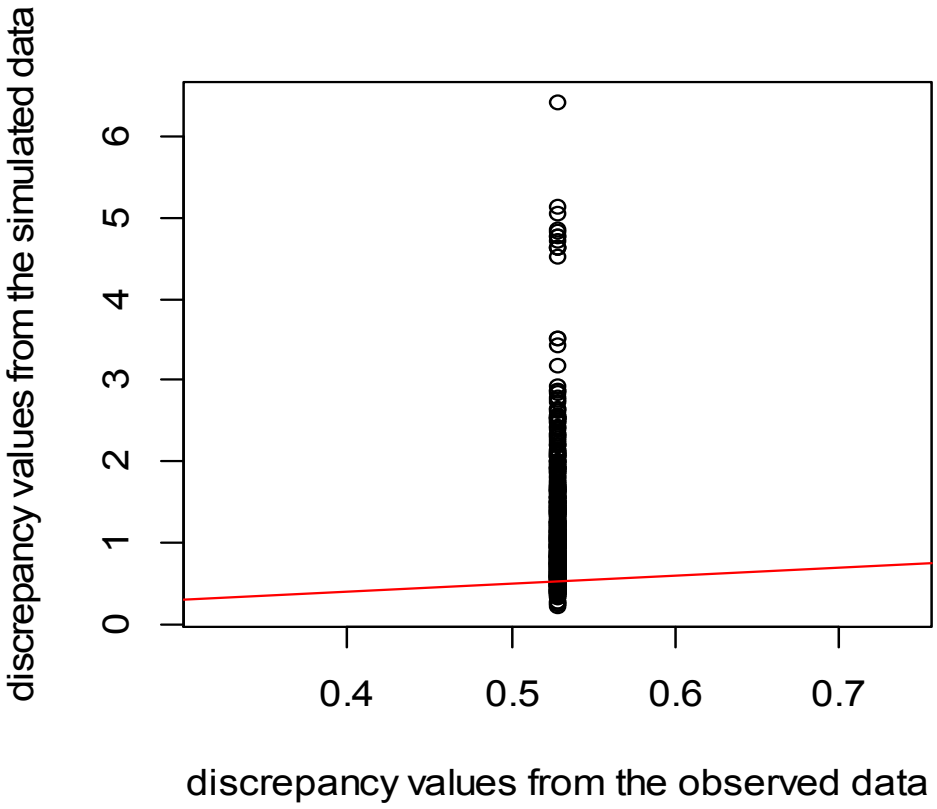

Figure 7.6: Scatter-plot of $D(f_j; \widehat{N}_j, \hat{p}_{ji})$ vs $D(f_{ji}; \widehat{N}_i, \hat{p}_{\mathrm{ji}})$.

We calibrate the observed p-values based on 100 simulated "observed" data sets, as described in Section 7.3.1.. Fig. 7.7 provides a boxplot of the simulated p-values from the original approach.

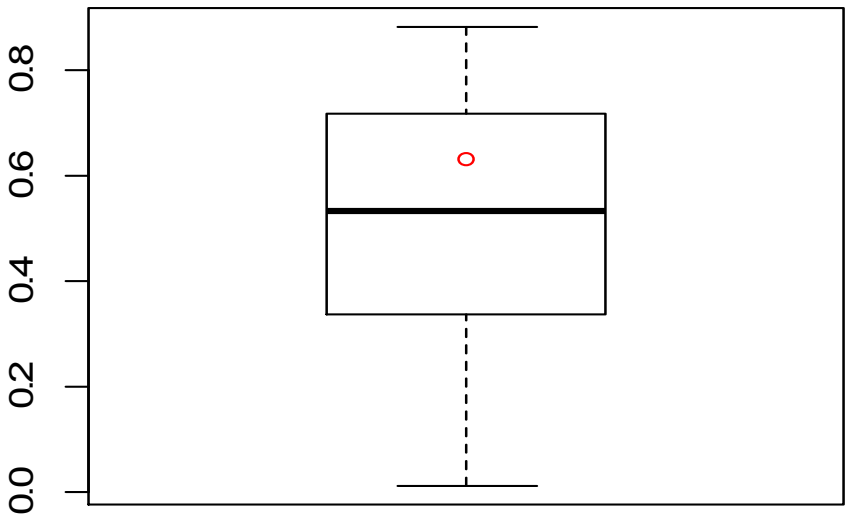

Figure 7.7: Boxplot of the simulated p-values from the original approach. The circle indicates the location of the p-value from the observed data.

We can see the observed p-value (0.634) is above the black line so we could say that the model fits the data well and the M_t is a good model for our data. We repeat the procedure for the new variant where the $\boldsymbol{x}_i$ data sets are generated from the maximum likelihood estimates.

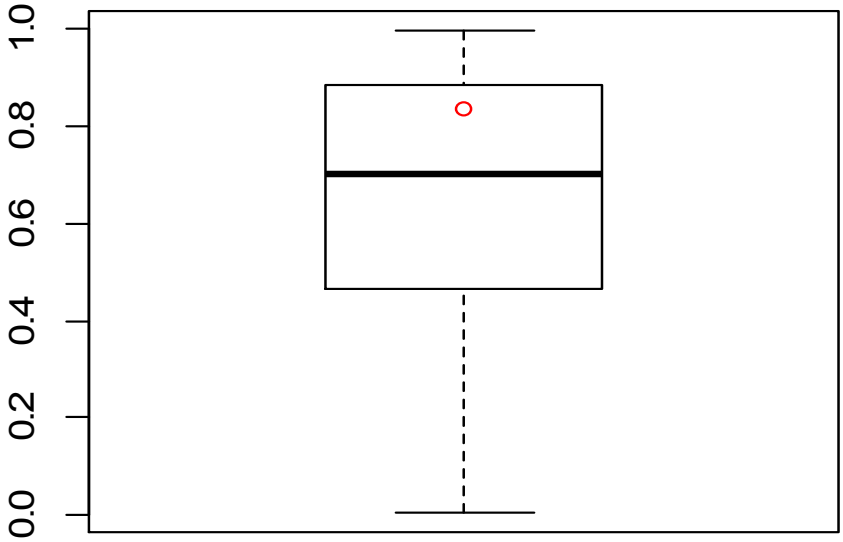

Figure 7.8: Boxplot of the simulated p-values from the variant. The circle indicates the location of the p-value from the observed data.

We can see that the observed p-value (0.872) is almost on the black line in the box-plot so we could say that our model fits very well our data.

APPENDIX

```
#Fecundability example#.
dat<-c(rep(1:13,c(198,107,55,38,18,22,7,9,5,3,6,6,12)))
data<-cbind(1:13,c(198,107,55,38,18,22,7,9,5,3,6,6,12))
satlogl<-function(x,y){
n<-nrow(x)
sum(x[1:n,2]*log(x[1:n,2]/length(y)))
}
#geom#
#simple loglikelihood#
geom1<-function(x,p){
p*(1-p)^(x-1)
}
loglik<-function(x,p){
-sum(log(geom1(x,p)))
}
a1<-optim(p=0.5,loglik,x=dat,method="BFGS",hessian=TRUE)
se1<-sqrt(diag(solve(a1$hessian)))
expec1<-geom1(1:12,a1$par)*486
expec1<-c(expec1,486-sum(expec1))
chisqstat1<-sum((data[1:9,2]-expec1[1:9])^2/(expec1[1:9]))
pval1<-pchisq(chisqstat1,7,lower.tail=FALSE)
slog<-satlogl(data,dat)
Dev1<-(-2*(-a1$value-slog))
#multinomial likelihood#
mulik<-function(x,p){
n<-nrow(x)
a<-sum(x[1:n-1,2]*log(geom1(x[1:n-1,1],p)))
b<-(x[n,2]*log(1-sum(geom1(x[1:n-1,1],p))))
-a-b
}
b1<-optim(p=0.5,mulik,x=data,method="BFGS",hessian=TRUE)
se2<-sqrt(diag(solve(b1$hessian)))
```

```
expec2<-geom1(1:12,b1$par)*486
expec2<-c(expec2,486-sum(expec2))
chisqstat2<-sum((data[1:9,2]-expec2[1:9])^2/(expec2[1:9]))
pval2<-pchisq(chisqstat2,11,lower.tail=FALSE)
Dev2<-(-2*(-b1$value-slog))
#Boostrap#
R1<-rep(0,500)
R2<-rep(0,500)
R3<-rep(0,500)
R4<-rep(0,500)
dens1<-rep(0,13)
for(i in 1:500){
bsdat<-rgeom(486,a1$par)+1
for(j in 1:12){
if (length(bsdat[bsdat[]==j]>0))
dens1[j]<-length(bsdat[bsdat[]==j])
else{
dens1[j]<-1
}
}
dens1[13]<-length(bsdat[bsdat[]>=13])
bsdata<-cbind(1:13,dens1)
bsmle<-optim(p=0.5,mulik,x=bsdata,method="BFGS",hessian=TRUE)
slog<-satlogl(bsdata,bsdat)
R1[i]<-bsmle$par
R2[i]<-bsmle$value
R3[i]<-bsmle$hessian
R4[i]<-slog
}
DEV1<-(-2*(-R2-R4))
#histogram#
hist(DEV1,xlab="Deviances",main=paste("HistogramofDeviaces"),xlim=range(0,50))
abline(v=Dev2,col="18")
```

```
#betageometric#
#simple loglikelihood#
marg<-function(x,ab){
(beta(ab[1]+1,x+ab[2]-1))/(beta(ab[1],ab[2]))
}
slik<-function(x,ab){
-sum(log(marg(x,ab)))
}
a2<-optim(c(2,3),slik,x=dat,method="BFGS",hessian=TRUE)
se4<-sqrt(diag(solve(a2$hessian)))
expec4<-marg(1:12,a2$par)*486
expec4<-c(expec4,486-sum(expec4))
chisqstat4<-sum((data[1:9,2]-expec4[1:9])^2/(expec4[1:9]))
pval4<-pchisq(chisqstat4,11,lower.tail=FALSE)
slog<-satlogl(data,dat)
Dev3<-(-2*(-a2$value-slog))
#multinomial likelihood#
mulm<-function(x,ab){
n<-nrow(x)
lm1<-(sum(x[1:n-1,2]*log(marg(x[1:n-1,1],ab))))
lm2<-(x[n,2]*log(1-sum(marg(x[1:n-1,1],ab))))
-lm1-lm2
}
b2<-optim(c(2,3),mulm,x=data[,1:2],method="BFGS",hessian=TRUE)
se3<-sqrt(diag(solve(b2$hessian)))
expec3<-marg(1:12,b2$par)*486
expec3<-c(expec3,486-sum(expec3))
chisqstat3<-sum((data[1:9,2]-expec3[1:9])^2/(expec3[1:9]))
pval3<-pchisq(chisqstat3,7,lower.tail=FALSE)
slog<-satlogl(data,dat)
Dev4<-(-2*(-b2$value-slog))
#Boostrap#
K1<-rep(0,500)
```

```
K2<-rep(0,500)
K3<-rep(0,500)
K4<-rep(0,500)
dens1<-rep(0,13)
for(i in 1:500){
bsdat1<-rbetageom(486,a2$par[1],a2$par[2])+1
for(j in 1:12){
if (length(bsdat1[bsdat1[]==j]>0))
dens1[j]<-length(bsdat1[bsdat1[]==j])
else{
dens1[j]<-1
}
}
dens1[13]<-length(bsdat1[bsdat1[]>=13])
bsdata1<-cbind(1:13,dens1)
bsmle1<-optim(c(2,3),mulm,x=bsdata1,method="BFGS",hessian=TRUE)
slog1<-satlogl(bsdata1,bsdat1)
K1[i]<-bsmle1$par
K2[i]<-bsmle1$value
K3[i]<-bsmle1$hessian
K4[i]<-slog1
}
DEV2<-(-2*(-K2-K4))
#Histogram#
hist(DEV2,xlab="Deviances",main=paste("Histogram of Deviaces"))
abline(v=Dev4,col="18")
#Geometric with the new method#
dat<-c(rep(1:13,c(198,107,55,38,18,22,7,9,5,3,6,6,12)))
data<-cbind(1:13,c(198,107,55,38,18,22,7,9,5,3,6,6,12))
geom1<-function(x,p){
p*(1-p)^(x-1)
}
mulik<-function(x,p){
```

```
n<-nrow(x)
a<-sum(x[1:n-1,2]*log(geom1(x[1:n-1,1],p)))
b<-(x[n,2]*log(1-sum(geom1(x[1:n-1,1],p))))
-a-b
}
Dft<-function(x,y){
sum((sqrt(x)-sqrt(y))^2)
}
b1<-optim(p=0.5,mulik,x=data,method="BFGS",hessian=TRUE)
se2<-sqrt(diag(solve(b1$hessian)))
pi<-rnorm(500,b1$par,se2)
phats<-matrix(pi,500,1)
#a)  X's  generate by rgeom with p=MLE#
chihats1<-matrix(0,13,500)
for( i in 1:500){
chis1<-rgeom(486,b1$par)+1
dens1<-rep(0,13)
dens1[13]<-length(chis1[chis1[]>=13])
for(j in 1:12){
dens1[j]<-length(chis1[chis1[]==j])
}
chihats1[,i]<-dens1
}
expval<-matrix(0,13,500)
for(i in 1:500){
expec1<-geom1(1:12,b1$par)*486
expec1<-c(expec1,486-sum(expec1))
expval[,i]<-expec1
}
Dft<-function(x,y){
sum((sqrt(x)-sqrt(y))^2)
}
DFTa<-matrix(0,500,1)
```

```
for (i in 1:500){
disc<-Dft(data[1:13,2],expval[1:13,i])
DFTa[i,]<-disc
}
DFT1<-matrix(0,500,1)
for (i in 1:500){
disc<-Dft(chihats1[1:13,i],expval[1:13,i])
DFT1[i,]<-disc
}
sum(DFT1>DFTa)/500
plot(DFTa,DFT1,xlab="discrepancy values from the observed data",
ylab="discrepancy values from the simulated data")
abline(0,1)
#β) X's generate by rgeom with p=phats#
chihats2<-matrix(0,13,500)
for( i in 1:500){
chis2<-rgeom(486,phats[i,])+1
dens2<-rep(0,13)
dens2[13]<-length(chis2[chis2[]>=13])
for(j in 1:12){
dens2[j]<-length(chis2[chis2[]==j])
}
chihats2[,i]<-dens2
}
expval<-matrix(0,13,500)
for(i in 1:500){
expec1<-geom1(1:12,phats[i,])*486
expec1<-c(expec1,486-sum(expec1))
expval[,i]<-expec1
}
Dft<-function(x,y){
sum((sqrt(x)-sqrt(y))^2)
}
```

```
DFTb<-matrix(0,500,1)
for (i in 1:500){
disc<-Dft(data[1:13,2],expval[1:13,i])
DFTb[i,]<-disc
}
DFT2<-matrix(0,500,1)
for (i in 1:500){
disc<-Dft(chihats2[1:13,i],expval[1:13,i])
DFT2[i,]<-disc
}
sum(DFT2>DFTb)/500
plot(DFTb,DFT2,xlab="discrepancy values from the observed data",
ylab="discrepancy values from the simulated data")
abline(0,1)
#For Calibration we do as follow#
#a) X's generate by rgeom with p=MLE#
box1<-rep(0,100)
for( k in 1:100){
obs<-cbind(1:13,rep(0,13))
obs1<-rgeom(486,b1$par)+1
for( m in 1:13){
obs[m,2]<-length(obs1[obs1[]==m])
}
b1<-optim(p=0.5,mulik,x=obs,method="BFGS",hessian=TRUE)
se2<-sqrt(diag(solve(b1$hessian)))
pi<-rnorm(500,b1$par,se2)
phats<-matrix(pi,500,1)
expval<-matrix(0,13,500)
for(i in 1:500){
expec1<-geom1(1:12,b1$par)*486
expec1<-c(expec1,486-sum(expec1))
expval[,i]<-expec1
}
```


```
DFT0<-matrix(0,500,1)
for (i in 1:500){
disc<-Dft(obs[1:13,2],expval[1:13,i])
DFT0[i,]<-disc
}
chihats1<-matrix(0,13,500)
for( i in 1:500){
chis1<-rgeom(486,b1$par)+1
dens1<-rep(0,13)
dens1[13]<-length(chis1[chis1[]>=13])
for(j in 1:12){
dens1[j]<-length(chis1[chis1[]==j])
}
chihats1[,i]<-dens1
}
DFT3<-matrix(0,500,1)
for (i in 1:500){
disc<-Dft(chihats1[1:13,i],expval[1:13,i])
DFT3[i,]<-disc
}
Pc<-sum(DFT3>DFT0)/500
box1[k]<-Pc
}
boxplot(box1)
points((sum(DFT1>DFTa))/500,col=2)
#β) X's  generate by rgeom with p=phats#
box2<-rep(0,100)
for( k in 1:100){
obs<-cbind(1:13,rep(0,13))
obs1<-rgeom(486,b1$par)+1
for( m in 1:13){
obs[m,2]<-length(obs1[obs1[]==m])
}
```

```
b1<-optim(p=0.5,mulik,x=obs,method="BFGS",hessian=TRUE)
se2<-sqrt(diag(solve(b1$hessian)))
pi<-rnorm(500,b1$par,se2)
phats<-matrix(pi,500,1)
expval<-matrix(0,13,500)
for(i in 1:500){
expec1<-geom1(1:12,phats[i,])*486
expec1<-c(expec1,486-sum(expec1))
expval[,i]<-expec1
}
DFT0<-matrix(0,500,1)
for (i in 1:500){
disc<-Dft(obs[1:13,2],expval[1:13,i])
DFT0[i,]<-disc
}
chihats2<-matrix(0,13,500)
for( i in 1:500){
chis2<-rgeom(486,phats[i,])+1
dens2<-rep(0,13)
dens2[13]<-length(chis2[chis2[]>=13])
for(j in 1:12){
dens2[j]<-length(chis2[chis2[]==j])
}
chihats2[,i]<-dens2
}
DFT4<-matrix(0,500,1)
for (i in 1:500){
disc<-Dft(chihats2[1:13,i],expval[1:13,i])
DFT4[i,]<-disc
}
Pc<-sum(DFT4>DFT0)/500
box2[k]<-Pc
}
```

```
boxplot(box2)
points((sum(DFT2>DFTb))/500,col=2)
#Beta-Geometric with the new method#
dat<-c(rep(1:13,c(198,107,55,38,18,22,7,9,5,3,6,6,12)))
data<-cbind(1:13,c(198,107,55,38,18,22,7,9,5,3,6,6,12))
marg<-function(x,ab){
(beta(ab[1]+1,x+ab[2]-1))/(beta(ab[1],ab[2]))
}
mulm<-function(x,ab){
n<-nrow(x)
lm1<-(sum(x[1:n-1,2]*log(marg(x[1:n-1,1],ab))))
lm2<-(x[n,2]*log(1-sum(marg(x[1:n-1,1],ab))))
-lm1-lm2
}
Dft<-function(x,y){
sum((sqrt(x)-sqrt(y))^2)
}
b2<-optim(c(2,3),mulm,x=data[,1:2],method="BFGS",hessian=TRUE)
se3<-sqrt(diag(solve(b2$hessian)))
phats<-matrix(0,500,2)
for( i in 1:2){
pi<-rnorm(500,b2$par[i],se3[i])
phats[,i]<-pi
}
#a) X'ς  generated by rbetageom with a,b=MLE#
chihats1<-matrix(0,13,500)
for( i in 1:500){
chis1<-rbetageom(486,b2$par[1],b2$par[2])+1
dens1<-rep(0,13)
dens1[13]<-length(chis1[chis1[]>=13])
for(j in 1:12){
dens1[j]<-length(chis1[chis1[]==j])
}
```

```
chihats1[,i]<-dens1
}
expval<-matrix(0,13,500)
for(i in 1:500){
expec1<-marg(1:12,b2$par)*486
expec1<-c(expec1,486-sum(expec1))
expval[,i]<-expec1
}
DFTa<-matrix(0,500,1)
for (i in 1:500){
disc<-Dft(data[1:13,2],expval[1:13,i])
DFTa[i,]<-disc
}
DFT1<-matrix(0,500,1)
for (i in 1:500){
disc<-Dft(chihats1[1:13,i],expval[1:13,i])
DFT1[i,]<-disc
}
sum(DFT1>DFTa)/500
plot(DFTa,DFT1,xlab="discrepancy values from the observed data",
ylab="discrepancy values from the simulated data")
abline(0,1,col=18)
#β)X's  generated by  rbetageom with a=ahats,b=bhats, phats[ahats,bhats]#
chihats2<-matrix(0,13,500)
for( i in 1:500){
chis2<-rbetageom(486,phats[i,1],phats[i,2])+1
dens2<-rep(0,13)
dens2[13]<-length(chis2[chis2[]>=13])
for(j in 1:12){
dens2[j]<-length(chis2[chis2[]==j])
}
chihats2[,i]<-dens2
}
```

```
expval<-matrix(0,13,500)
for(i in 1:500){
expec1<-marg(1:12,phats[i,])*486
expec1<-c(expec1,486-sum(expec1))
expval[,i]<-expec1
}
DFTb<-matrix(0,500,1)
for (i in 1:500){
disc<-Dft(data[1:13,2],expval[1:13,i])
DFTb[i,]<-disc
}
DFT2<-matrix(0,500,1)
for (i in 1:500){
disc<-Dft(chihats2[1:13,i],expval[1:13,i])
DFT2[i,]<-disc
}
sum(DFT2>DFTb)/500
plot(DFTb,DFT2,xlab="discrepancy values from the observed data",
ylab="discrepancy values from the simulated data")
abline(0,1,col=18)
#For calibration we do as follow#
#a) X's generated by rbetageom with a,b=MLE#
box1<-rep(0,100)
for( k in 1:100){
obs<-cbind(1:13,rep(0,13))
obs1<-rbetageom(486,b2$par[1],b2$par[2])+1
for( m in 1:13){
obs[m,2]<-length(obs1[obs1[]==m])
}
b2<-optim(c(2,3),mulm,x=obs[,1:2],method="BFGS",hessian=TRUE)
se3<-sqrt(diag(solve(b2$hessian)))
phats<-matrix(0,500,2)
for( i in 1:2){
```

```
pi<-rnorm(500,b2$par[i],se3[i])
phats[,i]<-pi
}
expval<-matrix(0,13,500)
for(i in 1:500){
expec1<-marg(1:12,b2$par)*486
expec1<-c(expec1,486-sum(expec1))
expval[,i]<-expec1
}
DFT0<-matrix(0,500,1)
for (i in 1:500){
disc<-Dft(obs[1:13,2],expval[1:13,i])
DFT0[i,]<-disc
}
chihats1<-matrix(0,13,500)
for( i in 1:500){
chis1<-rbetageom(486,b2$par[1],b2$par[2])+1
dens1<-rep(0,13)
dens1[13]<-length(chis1[chis1[]>=13])
for(j in 1:12){
dens1[j]<-length(chis1[chis1[]==j])
}
chihats1[,i]<-dens1
}
DFT3<-matrix(0,500,1)
for (i in 1:500){
disc<-Dft(chihats1[1:13,i],expval[1:13,i])
DFT3[i,]<-disc
}
Pc<-sum(DFT3>DFT0)/500
box1[k]<-Pc
}
boxplot(box1)
```


```
points((sum(DFT1>DFTa))/500,col=2)
#β)X's  generated by rbetageom with a=ahats,b=bhats, phats[ahats,bhats]#
box2<-rep(0,100)
for( k in 1:100){
obs<-cbind(1:13,rep(0,13))
obs1<-rbetageom(486,b2$par[1],b2$par[2])+1
for( m in 1:13){
obs[m,2]<-length(obs1[obs1[]==m])
}
b2<-optim(c(2,3),mulm,x=obs[,1:2],method="BFGS",hessian=TRUE)
se3<-sqrt(diag(solve(b2$hessian)))
phats<-matrix(0,500,2)
for( i in 1:2){
pi<-rnorm(500,b2$par[i],se3[i])
phats[,i]<-pi
}
expval<-matrix(0,13,500)
for(i in 1:500){
expec1<-marg(1:12,phats[i,])*486
expec1<-c(expec1,486-sum(expec1))
expval[,i]<-expec1
}
DFT0<-matrix(0,500,1)
for (i in 1:500){
disc<-Dft(obs[1:13,2],expval[1:13,i])
DFT0[i,]<-disc
}
chihats2<-matrix(0,13,500)
for( i in 1:500){
chis2<-rbetageom(486,phats[i,1],phats[i,2])+1
dens2<-rep(0,13)
dens2[13]<-length(chis2[chis2[]>=13])
for(j in 1:12){
```

```
dens2[j]<-length(chis2[chis2[]==j])
}
chihats2[,i]<-dens2
}
DFT4<-matrix(0,500,1)
for (i in 1:500){
disc<-Dft(chihats2[1:13,i],expval[1:13,i])
DFT4[i,]<-disc
}
Pc<-sum(DFT4>DFT0)/500
box2[k]<-Pc
}
boxplot(box2)
points((sum(DFT2>DFTb))/500,col=2)
#geometric comparison between the bootstrap method and the new method#
dat<-c(rep(1:13,c(198,107,55,38,18,22,7,9,5,3,6,6,12)))
data<-cbind(1:13,c(198,107,55,38,18,22,7,9,5,3,6,6,12))
geom1<-function(x,p){
p*(1-p)^(x-1)
}
satlogl<-function(x,y){
n<-nrow(x)
sum(x[x[1:n,2]>0,2]*log(x[x[1:n,2]>0,2]/length(y)))
}
loglik<-function(x,p){
-sum(log(geom1(x,p)))
}
mulik<-function(x,p){
n<-nrow(x)
a<-sum(x[1:n-1,2]*log(geom1(x[1:n-1,1],p)))
b<-(x[n,2]*log(1-sum(geom1(x[1:n-1,1],p))))
-a-b
}
```

```
Dft<-function(x,y){
sum((sqrt(x)-sqrt(y))^2)
}
a1<-optim(p=0.5,loglik,x=dat,method="BFGS",hessian=TRUE)
b1<-optim(p=0.5,mulik,x=data,method="BFGS",hessian=TRUE)
se2<-sqrt(diag(solve(b1$hessian)))
pi<-rnorm(500,b1$par,se2)
phats<-matrix(pi,500,1)
slog1<-satlogl(data,dat)
Dev1<-(-2*(-a1$value-slog1))
Dev2<-(-2*(-b1$value-slog1))
#a)p=M.L.E.#
obsv<-matrix(0,13,100)
obsv1<-matrix(0,486,100)
for( k in 1:100){
obs<-cbind(1:13,rep(0,13))
obs1<-rgeom(486,b1$par)+1
for (m in 1:12){
obs[m,2]<-length(obs1[obs1[]==m])
}
obs[13,2]<-length(obs1[obs1[]>=13])
obsv[,k]<-obs[,2]
obsv1[,k]<-obs1
}
box1<-rep(0,100)
for (k in 1:100){
R1<-rep(0,500)
R2<-rep(0,500)
R3<-rep(0,500)
dens2<-rep(0,13)
a11<-optim(p=0.5,mulik,x=cbind(1:13,obsv[,k]),method="BFGS")
for(i in 1:500){
bsdat<-rgeom(486,a11$par)+1
```


```
for(j in 1:12){
dens2[j]<-length(bsdat[bsdat[]==j])
}
dens2[13]<-length(bsdat[bsdat[]>=13])
bsdata<-cbind(1:13,dens2)
bsmle<-optim(p=0.5,mulik,x=bsdata,method="BFGS")
slog<-satlogl(bsdata,bsdat)
R1[i]<-bsmle$par
R2[i]<-bsmle$value
R3[i]<-slog
}
DEV1a<-(-2*(-a11$value-satlogl(cbind(1:13,obsv[,k]),obsv1[,k])))
DEV1<-(-2*(-R2-R3))
Pcb<-sum(DEV1>DEV1a)/500
box1[k]<-Pcb
}
box2<-rep(0,100)
for( k in 1:100){
obsv11<-cbind(1:13,obsv[,k])
b11<-optim(p=0.5,mulik,x=obsv11,method="BFGS",hessian=TRUE)
se2<-sqrt(diag(solve(b11$hessian)))
pi<-rnorm(500,b11$par,se2)
phats<-matrix(pi,500,1)
expval<-matrix(0,13,500)
for(i in 1:500){
expec1<-geom1(1:12,b11$par)*486
expec1<-c(expec1,486-sum(expec1))
expval[,i]<-expec1
}
DFT0<-matrix(0,500,1)
for (i in 1:500){
disc<-Dft(obsv11[1:13,2],expval[1:13,i])
DFT0[i,]<-disc
```

```
}
chihats1<-matrix(0,13,500)
for( i in 1:500){
chis1<-rgeom(486,b11$par)+1
dens1<-rep(0,13)
dens1[13]<-length(chis1[chis1[]>=13])
for(j in 1:12){
dens1[j]<-length(chis1[chis1[]==j])
}
chihats1[,i]<-dens1
}
DFT3<-matrix(0,500,1)
for (i in 1:500){
disc<-Dft(chihats1[1:13,i],expval[1:13,i])
DFT3[i,]<-disc
}
Pc<-sum(DFT3>DFT0)/500
box2[k]<-Pc
}
plot(box1,box2,xlab="first method's p-values", ylab="second method's p-values")
#beta-geometric comparison between the bootstrap method and the new method#
dat<-c(rep(1:13,c(198,107,55,38,18,22,7,9,5,3,6,6,12)))
data<-cbind(1:13,c(198,107,55,38,18,22,7,9,5,3,6,6,12))
satlogl<-function(x,y){
n<-nrow(x)
sum(x[x[1:n,2]>0,2]*log(x[x[1:n,2]>0,2]/length(y)))
}
marg<-function(x,ab){
(beta(ab[1]+1,x+ab[2]-1))/(beta(ab[1],ab[2]))
}
mulm<-function(x,ab){
n<-nrow(x)
lm1<-(sum(x[1:n-1,2]*log(marg(x[1:n-1,1],ab))))
```

```
lm2<-(x[n,2]*log(1-sum(marg(x[1:n-1,1],ab))))
-lm1-lm2
}
slik<-function(x,ab){
-sum(log(marg(x,ab)))
}
Dft<-function(x,y){
sum((sqrt(x)-sqrt(y))^2)
}
b2<-optim(c(2,3),mulm,x=data[,1:2],method="BFGS",hessian=TRUE)
se3<-sqrt(diag(solve(b2$hessian)))
phats<-matrix(0,500,2)
for( i in 1:2){
pi<-rnorm(500,b2$par[i],se3[i])
phats[,i]<-pi
}
slog1<-satlogl(data,dat)
Dev4<-(-2*(-b2$value-slog1))
#For calibration we do as follow#
#a) X's generated by rbetageom with a,b=MLE#
obsv<-matrix(0,13,100)
obsv1<-matrix(0,486,100)
for( k in 1:100){
obs<-cbind(1:13,rep(0,13))
obs1<-rbetageom(486,b2$par[1],b2$par[2])+1
for (m in 1:12){
obs[m,2]<-length(obs1[obs1[]==m])
}
obs[13,2]<-length(obs1[obs1[]>=13])
obsv[,k]<-obs[,2]
obsv1[,k]<-obs1
}
box1<-rep(0,100)
```

```
for (k in 1:100){
K1<-rep(0,500)
K2<-rep(0,500)
K3<-rep(0,500)
dens2<-rep(0,13)
b111<-optim(c(2,3),mulm,x=cbind(1:13,obsv[,k]),method="BFGS")
for(i in 1:500){
bsdat1<-rbetageom(486,b111$par[1],b111$par[2])+1
for(j in 1:12){
dens2[j]<-length(bsdat1[bsdat1[]==j])
}
dens2[13]<-length(bsdat1[bsdat1[]>=13])
bsdata1<-cbind(1:13,dens2)
bsmle1<-optim(c(2,3),mulm,x=bsdata1,method="BFGS")
slog<-satlogl(bsdata1,bsdat1)
K1[i]<-bsmle1$par
K2[i]<-bsmle1$value
K3[i]<-slog
}
DEV2b<-(-2*(-b111$value-satlogl(cbind(1:13,obsv[,k]),obsv1[,k])))
DEV2<-(-2*(-K2-K3))
Pcb<-sum(DEV2>DEV2b)/500
box1[k]<-Pcb
}
box2<-rep(0,100)
for( k in 1:100){
obs12<-cbind(1:13,obsv[,k])
b22<-optim(c(2,3),mulm,x=obs12,method="BFGS",hessian=TRUE)
se3<-sqrt(diag(solve(b22$hessian)))
phats<-matrix(0,500,2)
for( i in 1:2){
pi<-rnorm(500,b22$par[i],se3[i])
phats[,i]<-pi
```

```
}
expval<-matrix(0,13,500)
for(i in 1:500){
expec1<-marg(1:12,b22$par)*486
expec1<-c(expec1,486-sum(expec1))
expval[,i]<-expec1
}
DFT0<-matrix(0,500,1)
for (i in 1:500){
disc<-Dft(obs12[1:13,2],expval[1:13,i])
DFT0[i,]<-disc
}
chihats1<-matrix(0,13,500)
for( i in 1:500){
chis1<-rbetageom(486,b22$par[1],b22$par[2])+1
dens1<-rep(0,13)
dens1[13]<-length(chis1[chis1[]>=13])
for(j in 1:12){
dens1[j]<-length(chis1[chis1[]==j])
}
chihats1[,i]<-dens1
}
DFT3<-matrix(0,500,1)
for (i in 1:500){
disc<-Dft(chihats1[1:13,i],expval[1:13,i])
DFT3[i,]<-disc
}
Pc<-sum(DFT3>DFT0)/500
box2[k]<-Pc
}
plot(box1,box2)
#M0 model#
dat<-c(16,11,15,14,14,18)
```

```
mulik<-function(x,r,Np){
a<-lfactorial(Np[1])-lfactorial(Np[1]-r)
b<-(sum(x))*log(Np[2])+(6*Np[1]-sum(x))*log(1-Np[2])
-a-b
}
Dft<-function(x,y){
sum((sqrt(x)-sqrt(y))^2)
}
b1<-optim(c(55,0.3),mulik,x=dat,r=47,method="BFGS",hessian=TRUE)
mu<-c(b1$par[1],b1$par[2])
Nphats<-mvrnorm(500,mu,solve(b1$hessian))
#a) X's generated by rbinom with p=MLE,N=MLE#
chihats1<-matrix(0,6,500)
r1<-matrix(0,1,500)
for(i in 1:500){
M<-matrix(rbinom(round(b1$par[1])*6,1,b1$par[2]),nrow=round(b1$par[1]),ncol=6)
indx<-apply(M,1,sum)>0
m<-M[indx,]
m0<-apply(m,2,sum)
r0<-nrow(m)
r1[,i]<-r0
chihats1[,i]<-m0
}
expval<-matrix(0,6,500)
ex<-matrix(0,6,1)
for(i in 1:500){
for(j in 1:6){
expec1<-((b1$par[1])*b1$par[2])
ex[j,]<-expec1
}
expval[,i]<-expec1
}
Dft<-function(x,y){
```

```
sum((sqrt(x)-sqrt(y))^2)
}
DFTa<-matrix(0,500,1)
for (i in 1:500){
disc<-Dft(dat[1:6],expval[1:6,i])
DFTa[i,]<-disc
}
DFT1<-matrix(0,500,1)
for (i in 1:500){
disc<-Dft(chihats1[1:6,i],expval[1:6,i])
DFT1[i,]<-disc
}
sum(DFT1>DFTa)/500
plot(DFTa,DFT1,xlab="discrepancy values from the observed data",
ylab="discrepancy values from the simulated data")
abline(0,1,col=18)

#β) X's generated rbinom with N,p=Nphats#
chihats2<-matrix(0,6,500)
r2<-matrix(0,1,500)
for(i in 1:500){
M<-
matrix(rbinom(round(Nphats[i,1])*6,1,Nphats[i,2]),nrow=round(Nphats[i,1]),ncol=6)
indx<-apply(M,1,sum)>0
m<-M[indx,]
m0<-apply(m,2,sum)
r0<-nrow(m)
r2[,i]<-r0
chihats2[,i]<-m0
}
expval<-matrix(0,6,500)
ex<-matrix(0,6,1)
for(i in 1:500){
```

```
for(j in 1:6){
expec1<-((Nphats[i,1])*Nphats[i,2])
ex[j,]<-expec1
}
expval[,i]<-expec1
}
Dft<-function(x,y){
sum((sqrt(x)-sqrt(y))^2)
}
DFTb<-matrix(0,500,1)
for (i in 1:500){
disc<-Dft(dat[1:6],expval[1:6,i])
DFTb[i,]<-disc
}
DFT2<-matrix(0,500,1)
for (i in 1:500){
disc<-Dft(chihats2[1:6,i],expval[1:6,i])
DFT2[i,]<-disc
}
sum(DFT2>DFTb)/500
plot(DFTb,DFT2,xlab="discrepancy values from the observed data",
ylab="discrepancy values from the simulated data")
abline(0,1,col=18)



#For calibration we as follow#

#a)X's generated by rbinom with p=MLE,N=MLE#

box1<-rep(0,100)
b2est<-matrix(0,100,2)
for( k in 1:100){
```

```
obs<-rep(0,6)
M<-matrix(rbinom(round(b1$par[1])*6,1,b1$par[2]),nrow=round(b1$par[1]),ncol=6)
indx<-apply(M,1,sum)>0
m<-M[indx,]
m0<-apply(m,2,sum)
r0<-nrow(m)
obs<-m0
b2<-optim(c(55,0.3),mulik,x=obs,r=r0,method="BFGS",hessian=TRUE)
mu<-c(b2$par[1],b2$par[2])
Nphats<-mvrnorm(500,mu,solve(b2$hessian))
expval<-matrix(0,6,500)
ex<-matrix(0,6,1)
for(i in 1:500){
for(j in 1:6){
expec1<-((b2$par[1])*b2$par[2])
ex[j,]<-expec1
}
expval[,i]<-expec1
}
DFT0<-matrix(0,500,1)
for (i in 1:500){
disc<-Dft(obs[1:6],expval[1:6,i])
DFT0[i,]<-disc
}
chihats1<-matrix(0,6,500)
r1<-matrix(0,1,500)
for(i in 1:500){
M<-matrix(rbinom(round(b2$par[1])*6,1,b2$par[2]),nrow=round(b2$par[1]),ncol=6)
indx<-apply(M,1,sum)>0
m<-M[indx,]
m0<-apply(m,2,sum)
r0<-nrow(m)
r1[,i]<-r0
```


```
chihats1[,i]<-m0
}
DFT3<-matrix(0,500,1)
for (i in 1:500){
disc<-Dft(chihats1[1:6,i],expval[1:6,i])
DFT3[i,]<-disc
}
Pc<-sum(DFT3>DFT0)/500
box1[k]<-Pc
b2est[k,1]<-b2$par[1]
b2est[k,2]<-b2$par[2]
}
hist(b2est[,1]) #histogram of Nhats#
hist(b2est[,2]) #histogram of phats#
boxplot(box1)
points((sum(DFT1>DFTa))/500,col=2)
#β)X's generated by rbinom with N=Nhats,p=phats#
box2<-rep(0,100)
b2est<-matrix(0,100,2)
for( k in 1:100){
obs<-rep(0,6)
M<-matrix(rbinom(round(b1$par[1])*6,1,b1$par[2]),nrow=round(b1$par[1]),ncol=6)
indx<-apply(M,1,sum)>0
m<-M[indx,]
m0<-apply(m,2,sum)
r0<-nrow(m)
obs<-m0
b2<-optim(c(56,0.3),mulik,x=obs,r=r0,method="BFGS",hessian=TRUE)
mu<-c(b2$par[1],b2$par[2])
Nphats<-mvrnorm(500,mu,solve(b2$hessian))
expval<-matrix(0,6,500)
ex<-matrix(0,6,1)
for(i in 1:500){
```

```
for(j in 1:6){
expec1<-((Nphats[i,1])*Nphats[i,2])
ex[j,]<-expec1
}
expval[,i]<-expec1
}
DFT0<-matrix(0,500,1)
for (i in 1:500){
disc<-Dft(obs[1:6],expval[1:6,i])
DFT0[i,]<-disc
}
chihats2<-matrix(0,6,500)
r2<-matrix(0,1,500)
for(i in 1:500){
M<-
matrix(rbinom(round(Nphats[i,1])*6,1,Nphats[i,2]),nrow=round(Nphats[i,1]),ncol=6)
indx<-apply(M,1,sum)>0
m<-M[indx,]
m0<-apply(m,2,sum)
r0<-nrow(m)
r2[,i]<-r0
chihats2[,i]<-m0
}
DFT4<-matrix(0,500,1)
for (i in 1:500){
disc<-Dft(chihats2[1:6,i],expval[1:6,i])
DFT4[i,]<-disc
}
Pc<-sum(DFT4>DFT0)/500
box2[k]<-Pc
b2est[k,1]<-b2$par[1]
b2est[k,2]<-b2$par[2]
}
```


```
hist(b2est[,1]) #histogram of Nhats#
hist(b2est[,2]) #histogram of phats#
boxplot(box2)
points((sum(DFT2>DFTb))/500,col=2)
#M_t model#
dat<-c(28,53,50,60,37)
r<-127
fi<-c(52,52,20,3,0)
mulik<-function(x,r,N){
a<-lfactorial(N)-lfactorial(N-r)+x[1]*log(x[1]/N)+(N-x[1])*log(1-(x[1]/N))
b<-x[2]*log(x[2]/N)+(N-x[2])*log(1-(x[2]/N))+x[3]*log(x[3]/N)+(N-x[3])*log(1-
(x[3]/N))
c<-x[4]*log(x[4]/N)+(N-x[4])*log(1-(x[4]/N))+x[5]*log(x[5]/N)+(N-x[5])*log(1-
(x[5]/N))
-a-b-c
}
Dft<-function(x,y){
sum((sqrt(x)-sqrt(y))^2)
}
b1<-optim(150,mulik,x=dat,r=127,method="BFGS",hessian=TRUE)
pi<-matrix(0,1,5)
for (i in 1:5){
pi[i]<-dat[i]/b1$par
}
Npih<-c(b1$par,pi)
pid<-function(x,N,p){
-(x/(p^2))-((N-x)/((1-p)^2))
}
Niv<-(1/b1$par)-(1/(b1$par-r))
piv<-pid(dat[1:5],b1$par,pi[1:5])
Npiv<-c(Niv,piv)
piN<-function(x){
-1/(1-x)
```

```
}
piNh<-piN(pi[1:5])
varpN<-matrix(0,6,6)
for(i in 1:6){
varpN[i,i]<-Npiv[i]
}
for(i in 1:5){
varpN[1,1+i]<-piNh[i]
varpN[1+i,1]<-piNh[i]
}
Nphats<-mvrnorm(500,Npih,solve(-varpN))
#a)X's generated by rbinom with p=MLE,N=MLE#
fchihats1<-matrix(0,5,500)
for(i in 1:500){
M<-matrix(0,nrow=round(b1$par),ncol=5)
fix<-rep(0,5)
for( j in 1:5){
M[,j]<-matrix(rbinom(round(b1$par)*5,1,pi[j]),nrow=round(b1$par),ncol=1)
}
indx<-apply(M,1,sum)>0
m<-M[indx,]
g<-apply(m,1,sum)
for (k in 1:5){
fix[k]<-sum(g==k)
}
fchihats1[,i]<-fix
}
fiexp<-matrix(0,5,500)
for (i in 1:500){
fiexp1<-(dpoibin(kk=1:5,pp=pi[1:5],wts=NULL)*(b1$par))
fiexp[,i]<-fiexp1
}
Dft<-function(x,y){
```

```
sum((sqrt(x)-sqrt(y))^2)
}
DFTa<-matrix(0,500,1)
for (i in 1:500){
disc<-Dft(fi[1:5],fiexp[1:5,i])
DFTa[i,]<-disc
}
DFT1<-matrix(0,500,1)
for (i in 1:500){
disc<-Dft(fchihats1[1:5,i],fiexp[1:5,i])
DFT1[i,]<-disc
}
sum(DFT1>DFTa)/500
plot(DFTa,DFT1,xlab="discrepancy values from the observed data",
ylab="discrepancy values from the simulated data")
abline(0,1,col=18)

#β)X's generated by rbinom with N,p=Nphats#
fchihats2<-matrix(0,5,500)
r2<-matrix(0,1,500)
for(i in 1:500){
fix<-rep(0,5)
M1<-
matrix(rbinom(round(Nphats[i,1]),1,Nphats[i,2]),nrow=round(Nphats[i,1]),ncol=1)
M2<-
matrix(rbinom(round(Nphats[i,1]),1,Nphats[i,3]),nrow=round(Nphats[i,1]),ncol=1)
M3<-
matrix(rbinom(round(Nphats[i,1]),1,Nphats[i,4]),nrow=round(Nphats[i,1]),ncol=1)
M4<-
matrix(rbinom(round(Nphats[i,1]),1,Nphats[i,5]),nrow=round(Nphats[i,1]),ncol=1)
M5<-
matrix(rbinom(round(Nphats[i,1]),1,Nphats[i,6]),nrow=round(Nphats[i,1]),ncol=1)
M<-cbind(M1,M2,M3,M4,M5)
```


```
indx<-apply(M,1,sum)>0
m<-M[indx,]
g<-apply(m,1,sum)
for (k in 1:5){
fix[k]<-sum(g==k)
}
fchihats2[,i]<-fix
}
fiexp<-matrix(0,5,500)
for (i in 1:500){
fiexp2<-(dpoibin(kk=1:5,pp=Nphats[i,2:5],wts=NULL)*(Nphats[i,1]))
fiexp[,i]<-fiexp2
}
Dft<-function(x,y){
sum((sqrt(x)-sqrt(y))^2)
}
DFTb<-matrix(0,500,1)
for (i in 1:500){
disc<-Dft(fi[1:5],fiexp[1:5,i])
DFTb[i,]<-disc
}
DFT2<-matrix(0,500,1)
for (i in 1:500){
disc<-Dft(fchihats2[1:5,i],fiexp[1:5,i])
DFT2[i,]<-disc
}
sum(DFT2>DFTb)/500
plot(DFTb,DFT2,xlab="discrepancy values from the observed data",
ylab="discrepancy values from the simulated data")
abline(0,1,col=18)
#For calibration we do as follow#
#a)X's generated by rbinom with p=MLE,N=MLE#
box1<-rep(0,100)
```

```
for( k in 1:100){
fio<-rep(0,5)
M<-matrix(0,nrow=round(b1$par),ncol=5)
for( j in 1:5){
M[,j]<-matrix(rbinom(round(b1$par)*5,1,pi[j]),nrow=round(b1$par),ncol=1)
}
indx<-apply(M,1,sum)>0
m<-M[indx,]
g<-apply(m,1,sum)
for (j in 1:5){
fio[j]<-sum(g==j)
}
m0<-apply(m,2,sum)
r0<-nrow(m)
b2<-optim(c(150),mulik,x=m0,r=r0,method="BFGS")
pif<-rep(0,5)
for (j in 1:5){
pif[j]<-m0[j]/b2$par
}
fiexp<-matrix(0,5,500)
for (i in 1:500){
fiexp1<-dpoibin(kk=1:5,pp=pif[1:5],wts=NULL)*(b2$par)
fiexp[,i]<-fiexp1
}
DFT0<-matrix(0,500,1)
for (i in 1:500){
disc<-Dft(fio[1:5],fiexp[1:5,i])
DFT0[i,]<-disc
}
fchihats1<-matrix(0,5,500)
for(i in 1:500){
M<-matrix(0,nrow=round(b2$par),ncol=5)
fix2<-rep(0,5)
```


```
for( j in 1:5){
M[,j]<-matrix(rbinom(round(b2$par)*5,1,pif[j]),nrow=round(b2$par),ncol=1)
}
indx<-apply(M,1,sum)>0
m<-M[indx,]
g<-apply(m,1,sum)
for (j in 1:5){
fix2[j]<-sum(g==j)
}
fchihats1[,i]<-fix2
}
DFT3<-matrix(0,500,1)
for (i in 1:500){
disc<-Dft(fchihats1[1:5,i],fiexp[1:5,i])
DFT3[i,]<-disc
}
Pc<-sum(DFT3>DFT0)/500
box1[k]<-Pc
}
boxplot(box1)
points((sum(DFT1>DFTa))/500,col=2)
#β)X's generated by rbinom with N=Nhats,p=phats#
box2<-rep(0,100)
for( k in 1:100){
fio<-rep(0,5)
M<-matrix(0,nrow=round(b1$par),ncol=5)
for( j in 1:5){
M[,j]<-matrix(rbinom(round(b1$par)*5,1,pi[j]),nrow=round(b1$par),ncol=1)
}
indx<-apply(M,1,sum)>0
m<-M[indx,]
g<-apply(m,1,sum)
for (j in 1:5){
```


```
fio[j]<-sum(g==j)
}
m0<-apply(m,2,sum)
r0<-nrow(m)
b2<-optim(c(150),mulik,x=m0,r=r0,method="BFGS",hessian=TRUE)
pif<-rep(0,5)
for (j in 1:5){
pif[j]<-m0[j]/b2$par
}
b2f<-c(b2$par,pif)
piv1<-pid(m0[1:5],b2$par,pif[1:5])
Niv1<-(1/b2$par)-(1/(b2$par-r0))
Npiv1<-c(Niv1,piv1)
piNh1<-piN(pif[1:5])
varpN1<-matrix(0,6,6)
for(j in 1:6){
varpN1[j,j]<-Npiv1[j]
}
for(j in 1:5){
varpN1[1,1+j]<-piNh1[j]
varpN1[1+j,1]<-piNh1[j]
}
Nphats<-mvrnorm(500,b2f,solve(-varpN1))
fiexp<-matrix(0,5,500)
for (i in 1:500){
fiexp1<-dpoibin(kk=1:5,pp=Nphats[i,2:6],wts=NULL)*(Nphats[i,1])
fiexp[,i]<-fiexp1
}
DFT0<-matrix(0,500,1)
for (i in 1:500){
disc<-Dft(fio[1:5],fiexp[1:5,i])
DFT0[i,]<-disc
}
```


```
fchihats2<-matrix(0,5,500)
r2<-matrix(0,1,500)
for(i in 1:500){
fix<-rep(0,5)
M1<-
matrix(rbinom(round(Nphats[i,1]),1,Nphats[i,2]),nrow=round(Nphats[i,1]),ncol=1)
M2<-
matrix(rbinom(round(Nphats[i,1]),1,Nphats[i,3]),nrow=round(Nphats[i,1]),ncol=1)
M3<-
matrix(rbinom(round(Nphats[i,1]),1,Nphats[i,4]),nrow=round(Nphats[i,1]),ncol=1)
M4<-
matrix(rbinom(round(Nphats[i,1]),1,Nphats[i,5]),nrow=round(Nphats[i,1]),ncol=1)
M5<-
matrix(rbinom(round(Nphats[i,1]),1,Nphats[i,6]),nrow=round(Nphats[i,1]),ncol=1)
M<-cbind(M1,M2,M3,M4,M5)
indx<-apply(M,1,sum)>0
m<-M[indx,]
g<-apply(m,1,sum)
for (j in 1:5){
fix[j]<-sum(g==j)
}
fchihats2[,i]<-fix
}
DFT4<-matrix(0,500,1)
for (i in 1:500){
disc<-Dft(fchihats2[1:5,i],fiexp[1:5,i])
DFT4[i,]<-disc
}
Pc<-sum(DFT4>DFT0)/500
box2[k]<-Pc
}
boxplot(box2)
points((sum(DFT2>DFTb))/500,col=2)
```

REFERENCES

Besbeas P. and Morgan B.J.T. *Goodness-of-fit of integrated population models using calibrated simulation.* Journal of methods in ecology and evolution 2014, 5, 1373-1382

Casella G. and Berger R.L. (2002). *Statistical inference.* Brooks/Cole

Chernick M.R. and LaBuddle R.A. (2011). *An introduction to bootstrap methods with applications to R.* Wiley

Damianou Ch. And Koutras M. (2003). *Introduction to statistics.* Symmetria, Athens.

Spector P. *Data manipulation with R.* Springer.

Verzani J. *Using R for introductory statistics.* Chapman & Hall/CRC

White G., Anderson D., Burnham K. and Otis D. (1982). *Capture-Recapture and Removal Methods for Sampling Closed Populations.* Los Alamos National Laboratory, New Mexico.