ESTIMATING THE MAXIMUM PROBABILITY OF CATEGORICAL CLASSES WITH APPLICATIONS TO BIOLOGICAL DIVERSITY MEASUREMENTS

A Thesis
Presented to
The Academic Faculty

by

Huy Huynh

In Partial Fulfillment
of the Requirements for the Degree
Doctor of Philosophy in the
School of Mathematics

Georgia Institute of Technology
August 2012

ESTIMATING THE MAXIMUM PROBABILITY OF CATEGORICAL CLASSES WITH APPLICATIONS TO BIOLOGICAL DIVERSITY MEASUREMENTS

Approved by:

Professor Christian Houdré, Advisor
School of Mathematics
Georgia Institute of Technology

Professor Karim Lounici
School of Mathematics
Georgia Institute of Technology

Professor Liang Peng
School of Mathematics
Georgia Institute of Technology

Professor Yajun Mei
School of Industrial and Systems Engineering
Georgia Institute of Technology

Professor Vladimir Koltchinskii
School of Mathematics
Georgia Institute of Technology

Date Approved: 1 July 2012

To my parents, Loc Huynh and Suong Nguyen;

my wife, Dung Trina La;

and to all of the family members and friends

who have supported and believed in me

in accomplishing this prestigious degree.

ACKNOWLEDGEMENTS

First and foremost, I would like to express my great gratitude to my advisor, Dr. Christian Houdré. From the day I arrived at Georgia Tech, Dr. Houdré has mentored me in taking classes, directed me in learning and doing research in probability and statistics, and supported me in pursuing my future career. I am and will always be grateful for his kind teaching and valuable advice. I would also like to thank Dr. Liang Peng for his help and contribution to my research work. In addition, I am thankful for Dr. Vladimir Koltchinskii, Dr. Karim Lounici, Dr. Yajun Mei and again Dr. Liang Peng for serving on my dissertation committee and providing helpful comments and feedback on my dissertation.

I am proud to be part of the School of Mathematics. The education, preparation and support I have received from the school has lead me to where I am today. In particular, I would like to extend my sincere gratitude to Dr. Luca Dieci for his passion and care for the graduate program and its students . Moreover, I am grateful for the training and experience I have acquired during my stay at Tech. This is especially true of Dr. Andrew, Klara and Rena who have guided my teaching efforts over the past years. As an international student from Vietnam, I believe the School of Mathematics provides the best support for its graduate teaching assistants who are non native English speakers, and Ms. Cathy Jacobson plays a big role in accomplishing this task. I also want to thank Annette, Sharon, Ms. Turner and Ms. Hinds whose help and kindness will always be appreciated.

There are many of my fellow graduate students who have made positive impacts

on me over the years. In particular, I want to thank Giang Do for being my closest friend who has always been there for me. I also want to say thank you to Allen and Jamie Hoffmeyer for their great friendship. Allen is the best officemate that I can ever ask for. His kind personality and willingness to help will certainly be remembered. Furthermore, I really appreciate Yun Gong, my amazing colleague and home-mate, for his help in numerous research discussions. Giang Do, Anh Tran, Thao Vuong and many other Vietnamese friends have left big marks on my life at Georgia Tech. The memories we shared during our time at Tech will stay with us no matter where we will be. In addition to friends at Tech, I would like to thank Soleil Kent for being an awesome friend who always puts her friend's interest at heart.

I owe so much to my parents, Loc Huynh and Suong Nguyen, who have been educating and supporting me in all of my life choices. My mom's sacrifice and my dad's determination in letting their youngest son study abroad eleven years ago started my journey in America. Moreover, I would like to thank my brother and sister, Bao Huynh and Ngoc Huynh, for their love and encouragement throughout my doctoral study. I also want to express my sincere gratitude to my uncle and aunt, Ninh and Ytho Nguyen, my cousins, Wayne Pham and Tramy Nguyen, and to many other relatives in Atlanta for their constant care and support for me during my undergraduate and graduate studies. In addition, I want to thank the Loftons, my host family, for giving me an opportunity to come to the U.S and loving me as their family member. Last but very importantly, I owe innumerable thanks to my wonderful wife, Trina. Her faith and belief in me over all these years is the ultimate foundation for anything that I have accomplished.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

SUMMARY

Consider a sequence of n independent and identically random variables taking values in a collection of m items $\{\alpha_1, \alpha_2, ..., \alpha_m\}$. Let p_j be the probability that a random variable is $\alpha_j, j = 1, ..., m$, and let $p_{max} = \max_{j=1,...,m} p_j$. We are interested in estimating p_{max} when n and m approach infinity. In doing so, two estimators are studied: a simple estimator associated with the maximum of the multinomial distribution $(n, p_1, p_2, ..., p_m)$ and a non-linear estimator associated with the length of the longest increasing subsequence, LI_n, of the sequence of n random variables above. In both cases, the limiting distribution of the estimators as n and m approach infinity simultaneously is obtained. For the simple estimator, with appropriate assumptions and a natural standardization, the limiting law is shown to be a Gumbel distribution, while the asymptotic distribution of the length of the longest increasing subsequence $LI_n(m)$ is shown to be a Tracy-Widom distribution. Next, the confidence intervals for p_{max} are constructed using the two estimators. In investigating the sufficiency of the estimation methods, we compare the mean square error of the estimators. The bias corrected estimators associated with the two approaches are also compared. The problem of estimating the multiplicity of p_{max} is then studied. Finally, we discuss applications of estimating the maximum probability in biological diversity. In particular, the Berger-Parker index, a commonly used index to measure biological diversity, can be estimated using the results obtained above. These estimation provide a more accurate diversity comparison among communities with growing number individuals and species.

CHAPTER I

A SIMPLE ESTIMATOR

1.1 Introduction

Let $R_1, R_2, ..., R_n, ...$ be a sequence of iid random variables taking values in an alphabet $\{\alpha_1, \alpha_2, ..., \alpha_m\}$. Let $p_r = \mathbb{P}(R_1 = \alpha_r), 1 \leq r \leq m, \sum_{r=1}^{m} p_r = 1,$ and $p_{max} = \max_{1 \leq r \leq m} p_r$. Let also $M = \{r = 1, ..., m : p_r = p_{max}\}$ and k^* be the cardinality of M.

Given a sequence $R_1, R_2, ..., R_n$ satisfying the assumption above; define

$$X_j = \sum_{i=1}^{n} \mathrm{I}_{\{R_i = \alpha_j\}},$$

and $X_{(m)} = \max_{j=1,...,m} X_j$.

For each $n = 1, 2, 3, ...,$ let $X_1(n), X_2(n), ..., X_m(n)$ have a joint multinomial distribution, with parameters $n, p_1, ..., p_m$, i.e., let

$$\mathbb{P}\left((X_1, ..., X_m) = (x_1, ..., x_m)\right) = \frac{n!}{\prod_{i=1}^{m} x_i!} \prod_{i=1}^{m} p_i^{x_i},$$

where the x_i are non-negative integers, $i = 1, ..., m, \sum_{i=1}^{m} x_i = n$, and where $p_i > 0$ for $i = 1, ..., m, \sum_{i=1}^{m} p_i = 1$.

As we shall discuss more in detail in Chapter 5, p_{max} relates to Berger-Parker index, a commonly used index to measure biological diversity of an ecological community. Hence, estimating p_{max}, as n and m approach infinity simuntaniously is of interest. In this first chapter, we introduce a simple estimator $\dfrac{X_{(m)}}{n}$ associated with the multinomial max $X_{(m)}$.

The maximum cell frequency in a multinomial distribution has been a popular topic for decades (in [11], [19], [10], [31], [2], [15], [17], [38], [29], [23], [16], [34], [18] and [9]). In particular, the maximum cell statistics has often been suggested for testing purposes such as for favorable numbers on a roulette wheel ([8]) or spikes detection ([21]). The multinomial maximum is described in various forms in different contexts; applications have occured in sequential clinical trials and in paranormal experiment ([7], [21]). The multinomial maximum can also be found in data oriented parsing ([5]), where it is used to estimate the probability of the most probable parse by sampling random derivations. In addition, the maximum cell frequency has been used as a measurement of diversity, a concept explained in [30] and [1] as follows: A traveler in a tropical forest notices a particular species and wishes to find some more like it. The maximum probability of appearance of any species, which can be interpreted as the probability associated with the appearance of the typical species, has an inverse relation with the diversity of the forest. More recently, the relationship among the maximum cell frequency in multinomial trials, the gamma distribution order statistics and the distribution used in randomization designs has been discussed in [33]. In particular, the distribution of the waiting time till one of the treatments reaches its quota is directly related to that of the multinomial maximum in the uniform case.

In Section 1.3, we find the asymptotic distribution of the multinomial maximum with an increasing number of classes. The uniform case has been dealt at great length in [18] as the limiting distribution of maximal occupancy of boxes. The procedure is described as follows. Suppose we want to allocate n particles in m cells given that each particle has an equal chance to be in any one of the cells. Let $X_j, j = 1, ..., m$, denote the number of particles in the j^{th} cell, and let $X_{(m)} = \max_{1 \le j \le m} X_j$. Suppose $m = m(n) \to \infty$, as $n \to \infty$, in such way that $m \log m/n \to 0$, as $n \to \infty$, then the asymptotic distribution of the standardized multinomial maximum is a Gumbel

distribution. Below we find the asymptotic distribution of the multinomial maximum with an increasing number of classes for the general case. The tools we use in the present paper are simpler than the ones developed in [18]. To obtain the result of Theorem 1.3.2 below, we also require more assumptions (See conditions (2)-(7) in Section 1.3 for detail). These assumptions imply in particular that we need to have $m^3/n \to 0$, as $n \to \infty$, in the uniform case.

1.2 Finite number of classes, i.e. $m < \infty$

1.2.1 For $k^* = 1$

Proposition 1.2.1.

If there is precisely one $p_j = p_{max}$, i.e., if $k^* = 1$, then:

$$\frac{X_{(m)} - np_{max}}{\sqrt{np_{max}(1-p_{max})}} \Longrightarrow N(0,1).$$

Proof. We can write

$$X_{(m)} = \sum_{i=1}^{n} \mathrm{I}_{\{R_i = \alpha_{p_{max}}\}},$$

where $\mathrm{I}_{\{R_i=\alpha_{p_{max}}\}} \sim Ber(p_{max})$ and $\alpha_{p_{max}}$ is the letter that satisfies $\mathbb{P}(R_i = \alpha_{p_{max}}) = p_{max}$. Thus by the Central Limit Theorem, we obtain the result claimed.

□

1.2.2 For $1 < k^* < m$

Proposition 1.2.2.

$$\frac{X_{(m)} - np_{max}}{\sqrt{np_{max}}} \Rightarrow T_{(k^*)},$$

where $T_{(k^*)} = \max_{j=1,\ldots,k^*} T_j$ and $(T_1, ..., T_{k^*}) \sim N(\boldsymbol{0}, \Sigma_{k^*,p_{max}})$, provided that

$$\Sigma_{k^*,p_{max}} = \begin{pmatrix} 1-p_{max} & -p_{max} & -p_{max} & \cdots & -p_{max} \\ -p_{max} & 1-p_{max} & -p_{max} & \cdots & \vdots \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ \vdots & \cdots & -p_{max} & 1-p_{max} & -p_{max} \\ -p_{max} & \cdots & \cdots & -p_{max} & 1-p_{max} \end{pmatrix}.$$

Proof. Observing that $\mathrm{I}_{\{R_i=\alpha_j\}} \sim Ber(p_j)$, by the Central Limit Theorem we get:

$$\left(\frac{X_j - np_j}{\sqrt{np_j}}\right)_{j=1}^{m} \Longrightarrow N(0, \Sigma_m), \tag{1}$$

$$\text{where } (\Sigma_m)_{i,j} = \frac{Cov(X_i, X_j)}{n\sqrt{p_i p_j}} = \begin{cases} 1 - p_j & \text{if } i = j, \\ -\sqrt{p_i p_j} & \text{if } i \neq j. \end{cases}$$

That is,

$$\Sigma_m = \begin{pmatrix} 1-p_1 & -\sqrt{p_1p_2} & -\sqrt{p_1p_3} & ... & -\sqrt{p_1p_m} \\ -\sqrt{p_2p_1} & 1-p_2 & -\sqrt{p_2p_3} & \dots & \vdots \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ \vdots & \cdots & -\sqrt{p_{m-1}p_{m-2}} & 1-p_{m-1} & -\sqrt{p_{m-1}p_m} \\ -\sqrt{p_mp_1} & \cdots & \cdots & -\sqrt{p_mp_{m-1}} & 1-p_m \end{pmatrix}.$$

Indeed, we can compute $Cov(X_i, X_j)$. First, we see that

$$\mathbb{E}(X_j) = np_j, \quad \text{and } Var(X_j) = np_j(1-p_j), \text{then we get}$$

$$\mathbb{E}(X_iX_j) = \sum_{k=1}^{n}\sum_{l=1}^{n} \mathbb{E}\left(\mathrm{I}_{\{R_k=\alpha_i\}}\mathrm{I}_{\{R_l=\alpha_j\}}\right) = \sum_{k=1}^{n}\sum_{l\neq k} \mathbb{E}\left(\mathrm{I}_{\{X_k=\alpha_i\}}\right)\mathbb{E}\left(\mathrm{I}_{\{X_l=\alpha_j\}}\right) = n(n-1)p_ip_j.$$

Hence,

$$Cov(X_i, X_j) = \mathbb{E}(X_iX_j) - \mathbb{E}(X_i)\mathbb{E}(X_j) = -np_ip_j$$

Now,

$$\mathbb{P}\left(\frac{X_{(m)} - np_{max}}{\sqrt{np_{max}}} \leq x\right)$$

$$= \mathbb{P}\left(X_{(m)} \leq np_{max} + x\sqrt{np_{max}}\right)$$

$$= \mathbb{P}\left(X_j \leq np_{max} + x\sqrt{np_{max}}, j = 1, ..., m\right)$$

$$= \mathbb{P}\left(\frac{X_j - np_j}{\sqrt{np_j}} \leq \frac{\sqrt{n}(p_{max} - p_j)}{\sqrt{p_j}} + x\sqrt{\frac{p_{max}}{p_j}}, j = 1, ..., m \right).$$

Note that for all $x \in \mathbb{R}$, $\frac{\sqrt{n}(p_{max} - p_j)}{\sqrt{p_j}} + x\sqrt{\frac{p_{max}}{p_j}} = \begin{cases} x & \text{if } j \in M, \\ \to \infty \text{ as } n \to \infty & \text{otherwise.} \end{cases}$

So, as $n \to \infty$, by (1) we obtain

$$\frac{X_{(m)} - np_{max}}{\sqrt{np_{max}}} \Longrightarrow T_{(k^*)},$$

where $T_{(k^*)} = \max_{j=1,...,k^*} T_j$ and $(T_1, ..., T_{k^*}) \sim N(\mathbf{0}, \Sigma_{k^*, p_{max}})$.

□

1.2.3 For $k^* = m$

Proposition 1.2.3. *(Uniform case)*

If $k = m$, i.e. if $p_{max} = 1/m$, then

$$\frac{X_{(m)} - n/m}{\sqrt{n/m}} \Longrightarrow Z_{(m)} - \bar{Z}_m,$$

where $Z_{(m)} = \max_{j=1,...,m} Z_j$ and $\bar{Z}_m = \frac{1}{m}\sum_{j=1}^{m} Z_j$, provided that $Z_1, Z_2, ..., Z_m$ are iid $N(0, 1)$.

Proof. From Proposition 1.2.2, $k^* = m$ implies as $n \to \infty$,

$$\frac{X_{(m)} - n/m}{\sqrt{n/m}} \Longrightarrow T_{(m)},$$

where $T_{(m)} = \max_{j=1,...,m} T_j$ and $(T_1, ..., T_m) \sim N(\mathbf{0}, \Sigma_{m,1/m})$, provided that

$$\Sigma_{m,1/m} = \frac{1}{m}\begin{pmatrix} m-1 & -1 & -1 & ... & -1 \\ -1 & m-1 & -1 & \dots & \vdots \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ \vdots & \cdots & -1 & m-1 & -1 \\ -1 & \cdots & \cdots & -1 & m-1 \end{pmatrix}.$$

Now since $\Sigma_{m,1/m}\Sigma_{m,1/m} = \Sigma_{m,1/m}$, we can write

$$(T_1, ..., T_m)' = \Sigma_m (Z_1, ..., Z_m)',$$

where $(Z_1, ..., Z_m) \sim N(0, \mathbf{I}_m)$, i.e., $Z_1, ..., Z_m \; iid \; N(0,1)$.

Equivalently,

$$T_j = \left(Z_j - \frac{1}{m} \sum_{i=1}^{m} Z_i \right), j = 1, ..., m.$$

Thus

$$\frac{X_{(m)} - n/m}{\sqrt{n/m}} \Longrightarrow \left(\max_{1 \le j \le m} Z_j - \frac{1}{m} \sum_{i=1}^{m} Z_i \right).$$

□

1.3 Increasing number of classes, i.e., $m = m(n) \to \infty$

In this section, we particularly focus on the case where $k^* = k^*(n) \to \infty$ as $n \to \infty$. As for the cases $k^* = 1$ and $k^* < \infty$, the result is the same as in Propositions 1.2.1 and 1.2.2, respectively.

Recall for each $n = 1, 2, 3, ...$, let $X_1(n), X_2(n), ..., X_{m(n)}(n)$ have a joint multinomial distribution, with parameters $n, p_1(n), ..., p_{m(n)}(n)$, i.e., let

$$\mathbb{P}\left((X_1, ..., X_m) = (x_1, ..., x_m)\right) = \frac{n!}{\prod_{i=1}^{m} x_i!} \prod_{i=1}^{m} p_i^{x_i},$$

where the x_i are non-negative integers, $i = 1, ..., m(n), \sum_{i=1}^{m(n)} x_i = n$, and where $p_i(n) > 0$ for $i = 1, ..., m(n), \sum_{i=1}^{m(n)} p_i(n) = 1$. Let $p_{max}(n) = \max_{1 \le j \le m} p_j(n)$, let $M(n) = \{j = 1, 2, ..., m : p_j(n) = p_{max}(n)\}$ and let $k^*(n)$ be the cardinality of $M(n)$. Also, for any $x \in \mathbb{R}$, let $c_{jn}(x) = \left(\frac{x}{a_{k^*}} + b_{k^*} \right) \frac{\sqrt{p_{max}}}{\sqrt{p_j}} + \sqrt{n} \frac{(p_{max} - p_j)}{\sqrt{p_j}}, j = 1, ..., m.$

Thoughout this section, we make the following assumptions:

$$m(n) \to \infty, k^*(n) \to \infty \text{ and } p_{max}(n) \to 0 \text{ as } n \to \infty. \tag{2}$$

$$\text{For some } \Delta \in (0,1), \min_{1 \le j \le m(n)} [1 - p_j(n)] > \Delta. \tag{3}$$

$$\sum_{j=1}^{m(n)} \frac{1}{\sqrt{np_j(n)}} \to 0 \text{ as } n \to \infty. \tag{4}$$

$$\frac{m(n)}{\sqrt{np_m(n)}} \to 0 \text{ as } n \to \infty. \tag{5}$$

$$\text{If } M^c(n) \neq \emptyset, \frac{1}{\sqrt{p_m(n)}} \sum_{j \in M^c} \frac{1}{c_{jn}(x)} e^{-c_{jn}^2(x)/2} \to 0 \text{ as } n \to \infty. \tag{6}$$

$$\text{For each } x \in \mathbb{R}, c_{mn}(x) \to \infty \text{ as } n \to \infty. \tag{7}$$

Above, the assumption (2) indicates that the number of classes $m(n)$ increases to infinity and the multiplicity of p_{max} also grows to infinity. The assumptions (3)- (5) come from [37]. Finally, conditions (6) and (7) are used in the proof of Lemma 1.3.1 and Theorem 1.3.2, respectively.

From assumptions (2)-(5) and the fact that $k^*(n)p_{max}(n) < 1$, $p_{max}(n)$ must also satisfy the following conditions:

$$\text{For some } \Delta \in (0,1), p_{max}(n) < \min\left\{\frac{1}{k^*(n)}, 1 - \Delta\right\}, \tag{8}$$

$$\lim_{n \to \infty} \frac{k^*(n)}{\sqrt{np_{max}(n)}} = 0. \tag{9}$$

Often, to simplify notation, we write p_{max}, p_j, k^*, m and M for $p_{max}(n), p_j(n), k^*(n), m(n)$ and $M(n)$.

Lemma 1.3.1. *For each $n = 1, 2, 3, ...$, let $X_1(n), X_2(n), ..., X_m(n)$ have a joint multinomial distribution with parameters $n, p_1(n), ..., p_m(n)$. Let $X_{(m-1)}$ be the maximum of the first $m-1$ elements, i.e., $X_{(m-1)} = \max_{1 \le j \le m-1} X_j$.*

Let the conditions (2)-(7) hold true. Then as $n \to \infty$,

$$a_{k^*}\left(\frac{X_{(m-1)} - np_{max}}{\sqrt{np_{max}}} - b_{k^*}\right) \Longrightarrow G,$$

where $a_{k^*} = (2\log k^*)^{1/2}$ *and* $b_{k^*} = (2\log k^*)^{1/2} - \frac{1}{2}(2\log k^*)^{-1/2}\left(\log\log k^* + \log 4\pi\right)$; *where* G *has a Gumbel distribution, i.e.,* $\mathbb{P}(G \leq x) = \exp\left(-\exp(-x)\right),\ x \in \mathbb{R}$ *and where* $\Longrightarrow$ *indicates convergence in distribution.*

Proof. Let $M'(n) := \{j = 1, 2, ..., m-1 : p_j(n) = p_{max}(n)\}$ and let $k'^*(n) :=$ the cardinality of M'.

Since $k^* - 1 \leq k'^* \leq k^*$ and $(M')^c \subset M^c$, the assumptions (2) and (6) imply that

$$k'^*(n) \to \infty \text{ as } n \to \infty. \tag{10}$$

$$\text{If } M'^c(n) \neq \emptyset, \text{ then } \frac{1}{\sqrt{p_m(n)}} \sum_{j \in M'^c} \frac{1}{c_{jn}(x)} e^{-c_{jn}^2(x)/2} \to 0 \text{ as } n \to \infty. \tag{11}$$

It is straightforward to verify that $\mathbb{E}(X_j) = np_j$, $Var(X_j) = np_j(1-p_j)$ and $Cov(X_i, X_j) = -np_ip_j,\ i, j = 1, ..., m, i \neq j$. Then,

$$\mathbb{P}\left(a_{k^*}\left(\frac{X_{(m-1)} - np_{max}}{\sqrt{np_{max}}} - b_{k^*}\right) \leq x\right)$$

$$= \mathbb{P}\left(X_j \leq \left(\frac{x}{a_{k^*}} + b_{k^*}\right)\sqrt{np_{max}} + np_{max}, j = 1, ..., m-1\right)$$

$$= \mathbb{P}\left(\frac{X_j - np_j}{\sqrt{np_j}} \leq \left(\frac{x}{a_{k^*}} + b_{k^*}\right)\frac{\sqrt{p_{max}}}{\sqrt{p_j}} + \sqrt{n}\frac{(p_{max} - p_j)}{\sqrt{p_j}}, j = 1, ..., m-1\right).$$

Set $Y_j = \dfrac{X_j - np_j}{\sqrt{np_j}}, j = 1, ..., m-1$. Then $\mathbb{E}(Y_j) = 0, Var(Y_j) = 1 - p_j$ and $Cov(Y_i, Y_j) = -\sqrt{p_ip_j},\ i, j = 1, ..., m-1, i \neq j$, and let

$$\Sigma_{m-1} = Cov(Y) = \begin{pmatrix} 1-p_1 & -\sqrt{p_1p_2} & -\sqrt{p_1p_3} & ... & -\sqrt{p_1p_m} \\ -\sqrt{p_2p_1} & 1-p_2 & -\sqrt{p_2p_3} & \dots & \vdots \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ \vdots & \cdots & -\sqrt{p_{m-2}p_{m-3}} & 1-p_{m-2} & -\sqrt{p_{m-2}p_{m-1}} \\ -\sqrt{p_{m-1}p_1} & \cdots & \cdots & -\sqrt{p_{m-1}p_{m-2}} & 1-p_{m-1} \end{pmatrix}.$$

Then

$$\mathbb{P}\left(a_{k^*}\left(\frac{X_{(m-1)} - np_{max}}{\sqrt{np_{max}}} - b_{k^*}\right) \leq x\right) = \mathbb{P}\left(Y_j \leq c_{jn}(x), j = 1, ..., m-1\right). \quad (12)$$

Let now $Z = (Z_1, Z_2, ..., Z_{m-1}) \sim N(0, \Sigma_{m-1})$ be jointly normal with density

$$f_{Z_1,...,Z_{m-1}}(z_1, ..., z_{m-1}) = \left(\frac{1}{2\pi}\right)^{(m-1)/2} p_m^{-1/2} \exp\left(-\frac{1}{2}\sum_{j=1}^{m-1} z_j^2 - \frac{1}{2p_m}(\sum_{j=1}^{m-1}\sqrt{p_j}z_j)^2\right).$$

Define $\bar{Z}_1, \bar{Z}_2, ..., \bar{Z}_m$ as the following functions of $Z_1, Z_2, ..., Z_{m-1}$. For $j = 1, ..., m-1, \bar{Z}_j$ is the closest value of Z_j which makes $np_{max} + \sqrt{np_{max}}\bar{Z}_j$ an integer. If there are two possible values for $\bar{Z}_j$, use the smaller one. $\bar{Z}_m$ is given by the identity

$$\sum_{j=1}^{m} \sqrt{p_j}\bar{Z}_j = 0.$$

From [37], we can write

$$\bar{Z}_j = Z_j + \frac{\theta_j}{2\sqrt{np_j}}, \text{where} \quad |\theta_j| \leq 1, j = 1, ..., m-1.$$

Also from [37], for any sequence $(A_n)_{n\geq 1}$, where for each n, A_n is a Borel subset of $\mathbb{R}^{m(n)}$, we obtain the following result

$$\left|\mathbb{P}\left((Y_1, ..., Y_{m-1}) \in A_n\right) - \mathbb{P}\left((\bar{Z}_1, ..., \bar{Z}_{m-1}) \in A_n\right)\right| \to 0, \text{ as } n \to \infty.$$

For each $x \in \mathbb{R}$, let $A_n = \Pi_{j=1}^{m}\left(-\infty, c_{jn}(x)\right)$, the result above implies as $n \to \infty$,

$$\left|\mathbb{P}\left(Y_j \leq c_{jn}, j = 1, ..., m-1\right) - \mathbb{P}\left(\bar{Z}_j \leq c_{jn}, j = 1, ..., m-1\right)\right| \to 0. \quad (13)$$

From the relation between $\bar{Z}_j$ and Z_j we have

$$a_{k^*}\left(\frac{\bar{Z}_j - \sqrt{n}\frac{p_{max}-p_j}{\sqrt{p_j}}}{\sqrt{p_{max}/p_j}} - b_{k^*}\right) \tag{14}$$

$$= a_{k^*}\left(\frac{Z_j - \sqrt{n}\frac{p_{max}-p_j}{\sqrt{p_j}}}{\sqrt{p_{max}/p_j}} - b_{k^*}\right) + \frac{a_{k^*}\theta_j}{2\sqrt{np_{max}}\sqrt{p_{max}/p_j}}.$$

Since $|\theta_j| \leq 1$ for $j = 1, ..., m-1$ and from (4), it is easily seen that as $n \to \infty$,

$$\sup_{1\leq j\leq m-1} \frac{a_{k^*}}{2\sqrt{np_{max}}\sqrt{p_{max}/p_j}}\theta_j \to 0. \tag{15}$$

Equation (14) and condition (15) together with Slutsky's lemma imply that for each $x \in \mathbb{R}$,

$$\lim_{n\to\infty} \left|\mathbb{P}\left(\bar{Z}_j \leq c_{jn}(x), j = 1, ..., m-1\right) - \mathbb{P}\left(Z_j \leq c_{jn}(x), j = 1, ..., m-1\right)\right|$$

$$= \lim_{n\to\infty} \left|\mathbb{P}\left(g(\bar{Z}_j) \leq x, j = 1, ..., m-1\right) - \mathbb{P}\left(g(Z_j) \leq x, j = 1, ..., m-1\right)\right|$$

$$= 0,$$

where $g(z) = a_{k^*}\left(\frac{z - \sqrt{n}\frac{p_{max}-p_j}{\sqrt{p_j}}}{\sqrt{p_{max}/p_j}} - b_{k^*}\right)$.

This result and (13) give as $n \to \infty$,

$$\left|\mathbb{P}\left(Y_j \leq c_{jn}(x), j = 1, ..., m-1\right) - \mathbb{P}\left(Z_j \leq c_{jn}(x), j = 1, ..., m-1\right)\right| \to 0. \tag{16}$$

Note that for $j \in M'$, $c_{jn}(x) = \frac{x}{a_{k^*}} + b_{k^*}$, and let

$$\Omega_n = \{(z_1, ..., z_{m-1}) \in \mathbb{R}^m : z_j \leq c_{jn}(x) \text{ for } j \in (M')^c\}.$$

Then

$$\left|\mathbb{P}\left(Z_j \leq c_{jn}(x), j = 1, ..., m-1\right) - \mathbb{P}\left(Z_j \leq \frac{x}{a_{k^*}} + b_{k^*}, j \in M'\right)\right|$$

$$= \int_{\Omega_n^c} f_{Z_1,...,Z_{m-1}}(z_1,...,z_{m-1})dz_1dz_2...dz_{m-1}$$

$$\leq \frac{1}{\sqrt{p_m}} \int_{\Omega_n^c} \left(\frac{1}{2\pi}\right)^{(m-1)/2} \exp\left(-\frac{1}{2}\sum_{j=1}^{m-1} z_j^2\right) dz_1dz_2...dz_{m-1}$$

$$\leq \frac{1}{\sqrt{p_m}} \sum_{j\in M'^c} \left(\int_{c_{jn}(x)}^{\infty} \frac{1}{2\pi}\exp(-z_j^2/2)dz_j\right)$$

$$\leq \frac{1}{\sqrt{p_m}} \sum_{j\in M'^c} \frac{1}{c_{jn}(x)} e^{-c_{jn}^2(x)/2},$$

where the last two inequalities follow respectively from Boole's inequality and a well known estimate on the standard normal survival function. Taking the limit as $n \to \infty$ and by (11) one obtains:

$$\left|\mathbb{P}\left(Z_j \leq c_{jn}(x), j = 1,...,m-1\right) - \mathbb{P}\left(Z_j \leq \frac{x}{a_{k^*}} + b_{k^*}, j \in M'\right)\right| \to 0. \tag{17}$$

Set $Z^* = (Z_1^*,...,Z_{k'^*}^*) = (Z_j, j \in M')$, then $Z^* \sim N(0, \Sigma_{k'^*})$ where

$$\Sigma_{k'^*} = \begin{pmatrix} 1-p_{max} & -p_{max} & -p_{max} & ... & -p_{max} \\ -p_{max} & 1-p_{max} & -p_{max} & \dots & \vdots \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ \vdots & \cdots & -p_{max} & 1-p_{max} & -p_{max} \\ -p_{max} & \cdots & \cdots & -p_{max} & 1-p_{max} \end{pmatrix}.$$

Then (16) and (17) imply that, as $n \to \infty$,

$$\left|\mathbb{P}\left(Y_j \leq c_{jn}(x), j = 1,...,m-1\right) - \mathbb{P}\left(Z_{(k'^*)}^* \leq \frac{x}{a_{k^*}} + b_{k^*}\right)\right| \to 0, \tag{18}$$

where $Z_{(k'^*)}^* = \max\limits_{1\leq j\leq k'^*} Z_j^*$.

Next, let $N = \frac{1}{\sqrt{1-p_{max}}} Z^*$, then N is a centered normal vector with covariance matrix

$$\begin{pmatrix} 1 & -\frac{p_{max}}{1-p_{max}} & -\frac{p_{max}}{1-p_{max}} & \dots & -\frac{p_{max}}{1-p_{max}} \\ -\frac{p_{max}}{1-p_{max}} & 1 & -\frac{p_{max}}{1-p_{max}} & \dots & \vdots \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ \vdots & \dots & -\frac{p_{max}}{1-p_{max}} & 1 & -\frac{p_{max}}{1-p_{max}} \\ -\frac{p_{max}}{1-p_{max}} & \dots & \dots & -\frac{p_{max}}{1-p_{max}} & 1 \end{pmatrix}.$$

Let $\rho_{k'^*} = \frac{2p_{max}}{1-p_{max}}$. Clearly, since $p_{max} \leq \frac{1}{k'^*}$, then $\rho_{k'^*} < 1$, for $k'^* \geq 4$ and by (10),

$$\rho_{k'^*} \log k'^* = \frac{2p_{max} \log k'^*}{1-p_{max}} < \frac{2 \log k'^*/k'^*}{1-1/k'^*} = \frac{2 \log k'^*}{k'^*-1} \to 0$$

as $k'^* \to \infty$.

Thus, by Theorem 6.2.1 in [20] with $m_i = m_i^* = 0$ and $|r_{ij}| = \frac{p_{max}}{1-p_{max}} < \rho_{|i-j|}$, $i, j = 1, ..., k'^*, i \neq j$, we get as $k'^* \to \infty$,

$$a_{k'^*} \left(N_{(k'^*)} - b_{k'^*} \right) \Longrightarrow G, \tag{19}$$

where $N_{(k'^*)} = \max_{1 \leq j \leq k'^*} N_j$ and G is a Gumbel distribution. Hence,

$$a_{k'^*} \left(Z^*_{(k'^*)} - b_{k'^*} \right) = \sqrt{1-p_{max}} a_{k'^*} \left(N_{(k'^*)} - b_{k'^*} \right) + a_{k'^*} b_{k'^*} \left(\sqrt{1-p_{max}} - 1 \right). \tag{20}$$

Note that as $k'^* \to \infty, \sqrt{1-p_{max}} \to 1$ and $a_{k'^*} b_{k'^*} \left(\sqrt{1-p_{max}} - 1 \right) \to 0$ since $p_{max} \leq 1/k'^*$ and

$$\left| a_{k'^*} b_{k'^*} \left(\sqrt{1-p_{max}} - 1 \right) \right| \leq 2 \log k'^* \frac{p_{max}}{\sqrt{1-p_{max}}+1} \leq \frac{2 \log k'^*}{k'^*} \frac{1}{\sqrt{1-p_{max}}+1} \to 0.$$

From (2), (19) and (20), the relations $a_{k'^*}/a_{k^*} \to 1$ and $a_{k^*}\left(b_{k^*} - b_{k'^*}\right) \to 0$ as $k^* \to \infty$, and Slutsky's lemma we get

$$a_{k^*}\left(Z^*_{(k'^*)} - b_{k^*}\right) \Longrightarrow G. \tag{21}$$

From (18) and (21) we obtain

$$\lim_{n\to\infty} \left|\mathbb{P}\left(Y_j \le c_{jn}(x), j = 1...m-1\right) - \exp\left(-\exp(x)\right)\right| = 0.$$

Recalling (12), we conclude that

$$\lim_{n\to\infty} \left|\mathbb{P}\left(a_{k^*}\left(\frac{X_{(m-1)} - np_{max}}{\sqrt{np_{max}}} - b_{k^*}\right) \le x\right) - \exp\left(-\exp(x)\right)\right| = 0.$$

□

Using Lemma 1.3.1, we now obtain the limiting distribution of the multinomial maximum with an increasing number of classes.

Theorem 1.3.2. *For each $n = 1, 2, 3, ...,$ let $X_1(n), X_2(n), ..., X_m(n)$ have a joint multinomial distribution with parameters $n, p_1(n), ..., p_m(n)$. Let $X_{(m)} = \max\limits_{1\le j\le m} X_j$.*

Let the conditions (2)-(7) hold true. Then as $n \to \infty$,

$$a_{k^*}\left(\frac{X_{(m)} - np_{max}}{\sqrt{np_{max}}} - b_{k^*}\right) \Longrightarrow G,$$

where $a_{k^*} = (2\log k^*)^{1/2}$ and $b_{k^*} = (2\log k^*)^{1/2} - \frac{1}{2}(2\log k^*)^{-1/2}\left(\log\log k^* + \log 4\pi\right)$.

Proof. Note that $\mathbb{P}\left(a_{k^*}\left(\frac{X_{(m)} - np_{max}}{\sqrt{np_{max}}} - b_{k^*}\right) \le x\right)$

$= \mathbb{P}\left(a_{k^*}\left(\frac{X_{(m-1)} - np_{max}}{\sqrt{np_{max}}} - b_{k^*}\right) \le x,\ \frac{X_m - np_m}{\sqrt{np_m}} \le c_{mn}(x)\right).$

The result above implies that, as $n \to \infty$,

$$\left| \mathbb{P}\left(a_{k^*}\left(\frac{X_{(m)} - np_{max}}{\sqrt{np_{max}}} - b_{k^*} \right) \leq x \right) - \mathbb{P}\left(a_{k^*}\left(\frac{X_{(m-1)} - np_{max}}{\sqrt{np_{max}}} - b_{k^*} \right) \leq x \right) \right|$$

$$\leq \mathbb{P}\left(\frac{X_m - np_m}{\sqrt{np_m}} \geq c_{mn}(x) \right) \to 0,$$

since $\dfrac{X_m - np_m}{\sqrt{np_m}} \Rightarrow N(0,1)$ and by condition (7). From Lemma 1.3.1, we then conclude that as $n \to \infty$,

$$a_{k^*}\left(\frac{X_{(m)} - np_{max}}{\sqrt{np_{max}}} - b_{k^*} \right) \Longrightarrow G.$$

□

We now specialize our main result to the uniform case where:

$$k^* = m,\ p_1(n) = p_2(n) = ... = p_m(n) = p_{max} = \frac{1}{m} \text{ and } M^c(n) = \emptyset.$$

Corollary 1.3.3. *For each $n = 1, 2, 3, ...,$ let $X_1(n), X_2(n), ..., X_{m(n)}(n)$ have a multinomial distribution with parameters n and $\left(\frac{1}{m}, \frac{1}{m}, ..., \frac{1}{m}\right)$, i.e.*

$$\mathbb{P}\left((X_1, ..., X_m) = (x_1, ..., x_m)\right) = \frac{n!}{\left(\prod_{j=1}^{m} x_j!\right)} \left(\frac{1}{m}\right)^n,$$

where x_i are non-negative integers, $i = 1, ..., m(n)$, $\sum_{i=1}^{m(n)} x_i = n$. Let $X_{(m)} = \max\limits_{1 \leq j \leq m} X_j$ and let $m = m(n) \to \infty$ as $n \to \infty$ in such a way that $\dfrac{m^3}{n} \to 0$. Then as $n \to \infty$,

$$a_m\left(\frac{X_{(m)} - \frac{n}{m}}{\sqrt{\frac{n}{m}}} - b_m \right) \Longrightarrow G,$$

where $a_m = (2\log m)^{1/2}$ and $b_m = (2\log m)^{1/2} - \frac{1}{2}(2\log m)^{-1/2}\left(\log\log m + \log 4\pi\right)$.

Proof. The assumptions (2)-(7) are satisfied. Therefore, the result follows from Theorem 1.3.2. □

CHAPTER II

A NON LINEAR ESTIMATOR

2.1 Introduction

Recall that $R_1, R_2, ..., R_n, ...$ is a sequence of iid random variables taking their values in a finite ordered alphabet $\{\alpha_1 < \alpha_2 < ... < \alpha_m\}$. Let $p_r = \mathbb{P}(R_1 = \alpha_r), 1 \leq r \leq m, \sum_{r=1}^{m} p_r = 1,$ and $p_{max} = \max_{1 \leq r \leq m}\{p_r\}$.

Now let LI_n be the length of the longest increasing subsequence of $R_1, R_2, ..., R_n$. To understand the definition of the LI_n better, let us consider the following example for $n = 10, m = 3$ with the alphabet $\{1, 2, 3\}$:

$$1 \quad 1 \quad 3 \quad 2 \quad 2 \quad 3 \quad 1 \quad 1 \quad 2 \quad 3.$$

A longest increasing subsequence is

$$\mathbf{1} \quad \mathbf{1} \quad 3 \quad \mathbf{2} \quad \mathbf{2} \quad \mathbf{3} \quad 1 \quad 1 \quad 2 \quad \mathbf{3}.$$

So the length of the longest increasing subsequence is $LI_n = 6$.

Note that the longest increasing subsequence is not unique. Here is another longest increasing subsequence of the example above:

$$\mathbf{1} \quad \mathbf{1} \quad 3 \quad 2 \quad 2 \quad 3 \quad \mathbf{1} \quad \mathbf{1} \quad \mathbf{2} \quad \mathbf{3}.$$

We introduce LI_n/n as an estimator for p_{max}. The study of the asymptotic distribution of LI_n is discussed. The result of the finite case can be found in [13] and the increasing number of classes case is studied in [6].

Motivating our investigation of LI_n in various probabilistic contexts is the classical problem of describing the length of the longest increasing subsequence of a random

permutation of the first n positive integers. The study of the asymptotic behavior of this quantity has enjoyed a rich history as 'Ulam's Problem'. There are good background information about the literature of the length of the longest increasing subsequence as well as the Ulam's Problem discussed in [13]. In this chapter, we also discuss the simulation study of LI_n and the max eigenvalue of the Gaussian Unitary Ensemble (GUE). This will give us a better idea about the distributions of the two random variables and their relation.

2.2 The length of the longest increasing subsequence

2.2.1 Combinatorics

Suppose a sequence $R_1, R_2, ...R_n, ...$ is described as above. Let a_k^r be the number of occurrences of α_r among $R_1, R_2, ...R_k, 1 \leq k \leq n$. Then the number of occurrences of $\alpha_r \in \{\alpha_1, ..., \alpha_m\}$ among $R_{k+1}, R_{k+2}, ...R_l$, where $1 \leq k < l \leq n$, is simply $a_l^r - a_k^r$.

The length of the longest increasing subsequence of $R_1, R_2, ..., R_n$ is then given by:

$$LI_n = \max_{0 \leq k_1 \leq ... \leq k_{m-1} \leq n} \left[(a_{k_1}^1 - a_0^1) + (a_{k_2}^2 - a_{k_1}^2) + ... + (a_n^m - a_{k_{m-1}}^m) \right],$$

i.e.,

$$LI_n = \max_{0 \leq k_1 \leq ... \leq k_{m-1} \leq n} [(a_{k_1}^1 - a_{k_1}^2) + (a_{k_2}^2 - a_{k_2}^3) + ... + (a_{k_{m-1}}^{m-1} - a_{k_{m-1}}^m) + a_n^m],$$

where $a_0^r = 0$. Now define $Z_i^r = \begin{cases} 1 & \text{,if } R_i = \alpha_r, \\ -1 & \text{, if } R_i = \alpha_{r+1}, \\ 0 & \text{, otherwise.} \end{cases}$

Let $S_k^r = \sum_{i=1}^k Z_i^r, k = 1, ..., n$ with $S_0^r = 0$. Then $S_k^r = a_k^r - a_k^{r+1}$and

$$LI_n = \frac{n}{m} - \frac{1}{m} \sum_{r=1}^{m-1} r S_n^r + \max_{0 \leq k_1 \leq ... \leq k_{m-1} \leq n} [S_{k_1}^1 + S_{k_2}^2 + ... + S_{k_{m-1}}^{m-1}]. \qquad (22)$$

2.2.2 Probabilistic development- The uniform case

Consider first the case in which $R_1, R_2, ..., R_n$ are i.i.d, with each letter drawn uniformly from $A = \{\alpha_1, ..., \alpha_m\}$. That is, $P(R_i = \alpha_r) = 1/m, 1 \leq i \leq n, 1 \leq r \leq m$. Then for each $r = 1, ..., m-1, \{Z_i^r\}$ forms i.i.d random variables with $\mathbb{E}[Z_i^r] = 0, \text{Var}(Z_i^r) = 2/m$, and so $\text{Var}(S_n^r) = 2n/m$. Define:

$$\hat{B}_n^r(t) = \frac{1}{\sqrt{\frac{2n}{m}}} S_{[nt]}^r + (nt - [nt]) \frac{1}{\sqrt{\frac{2n}{m}}} Z_{[nt]+1}^r$$

for $t \in [0, 1]$. Then for (22), we have:

$$\frac{LI_n - \frac{n}{m}}{\sqrt{\frac{2n}{m}}} = -\frac{1}{m} \sum_{i=1}^{m-1} i\hat{B}_n^i(1) + \max_{0 \leq t_1 \leq ... \leq t_{m-1} \leq 1} [\hat{B}_n^1(t_1) + ... + \hat{B}_n^{m-1}(t_{m-1})]. \tag{23}$$

By Donsker's theorem and Continuous Mapping Theorem for (m-1) dimensional random vector, (23) implies:

$$\frac{LI_n - \frac{n}{m}}{\sqrt{\frac{2n}{m}}} \Rightarrow -\frac{1}{m} \sum_{i=1}^{m-1} i\tilde{B}^i(1) + \max_{0 \leq t_1 \leq ... \leq t_{m-1} \leq 1} \sum_{i=1}^{m-1} \tilde{B}^i(t_i),$$

where the covariance structure of $(\tilde{B}^1(t), ...\tilde{B}^{m-1}(t))$ is

$$Cov(\tilde{B}^r(t), \tilde{B}^s(t)) = \begin{cases} t & \text{if } r = s, \\ 0 & \text{if } |r - s| \geq 2, \\ -\frac{t}{2} & \text{if } |r - s| = 1. \end{cases} \tag{24}$$

Proposition 2.2.1. *Let $R_1, R_2, ..., R_n$ be a sequence of i.i.d random variables drawn uniformly from the ordered finite alphabet $A = \{\alpha_1, ..., \alpha_m\}$. Then*

$$\frac{LI_n - \frac{n}{m}}{\sqrt{\frac{2n}{m}}} \Longrightarrow -\frac{1}{m} \sum_{i=1}^{m-1} i\tilde{B}^i(1) + \max_{0 \leq t_1 \leq ... \leq t_{m-1} \leq 1} \sum_{i=1}^{m-1} \tilde{B}^i(t_i),$$

where $(\tilde{B}^1(t), ...\tilde{B}^{m-1}(t))$ is an (m-1)-dimensional Brownian motion with covariance structure given in (24).

By letting $\tilde{B}^i(t) = \frac{1}{\sqrt{2}}(B^i(t) - B^{i+1}(t)), 1 \leq i \leq m-1$, *where* $(B^1(t), ..., B^m(t))$ *are standard Brownian motion, we can rewrite the result above as:*

$$\frac{LI_n - \frac{n}{m}}{\sqrt{\frac{n}{m}}} \Longrightarrow -\frac{1}{m}\sum_{i=1}^{m} B^i(1) + \max_{0=t_0\leq t_1\leq ...\leq t_{m-1}\leq t_m\leq 1} \sum_{i=1}^{m}[B^i(t_i) - B^i(t_{i-1})].$$

2.2.3 Probabilistic development- The non-uniform case

Theorem 2.2.2. *Let* $R_1, R_2, ..., R_n$ *be a sequence of i.i.d random variables drawn from the ordered finite alphabet* $A = \{\alpha_1, ..., \alpha_m\}$, *such that* $P(R_1 = \alpha_r) = p_r$, *for* $r = 1, ..., m$, *where* $0 < p_r < 1$ *and* $\sum_{r=1}^{m} p_r = 1$. *Then*

$$\frac{LI_n - p_{max}n}{\sqrt{n}} \Longrightarrow -\frac{1}{m}\sum_{i=1}^{m-1} i\sigma_i\tilde{B}^i(1) + \max_{0=t_0\leq t_1\leq ...\leq t_{m-1}\leq t_m=1, t_i=t_{i-1}, i\in I^*} \sum_{i=1}^{m-1}\sigma_i\tilde{B}^i(t_i),$$

where $p_{max} = \max_{1\leq r\leq r} p_r, \sigma_r = p_r + p_{r+1} - (p_r - p_{r+1})^2, M^c = \{r \in \{1, 2, ..., m\} : p_r < p_{max}\}$, *and where* $(\tilde{B}^1(t), ...\tilde{B}^{m-1}(t))$ *is an (m-1)-dimensional Brownian motion with covariance structure given in by:*

$$Cov(\tilde{B}^r(t), \tilde{B}^s(t)) = t.\begin{cases} 1 & \text{if } r = s, \\ -\frac{p_r+\mu_r\mu_s}{\sigma_r\sigma_s} & \text{if } s = r-1, \\ -\frac{p_s+\mu_r\mu_s}{\sigma_r\sigma_s} & \text{if } s = r+1, \\ -\frac{\mu_r\mu_s}{\sigma_r\sigma_s} & \text{if } |r-s| > 1, 1 \leq r, s \leq m-1, \end{cases}$$

with $\mu_r = p_r - p_{r+1}, 1 \leq r \leq m-1$.

There are 3 corollaries following Theorem 2.2.2:

Corollary 2.2.3. *If* $p_{max} = p_j$ *for precisely one* $j \in \{1, ..., m\}$, *then*

$$\frac{LI_n - p_{max}n}{\sqrt{n}} \Longrightarrow -\frac{1}{m}\sum_{i=1}^{m-1} i\sigma_i\tilde{B}^i(1) + \sum_{i=j}^{m-1}\sigma_i\tilde{B}^i(1),$$

where the last term is not present if $j = m$.

Corollary 2.2.4. *Let $p_{max} = p_{j_1} = p_{j_2} = ... = p_{j_{k^*}}$, for $1 \leq j_1 < j_2 < ... < j_{k^*} \leq m$ for some $1 \leq k^* \leq m-1$, and let $p_i < p_{max}$, otherwise. Then*

$$\frac{LI_n - p_{max}n}{\sqrt{n}} \Longrightarrow \sqrt{p_{max}(1-p_{max})} \max_{0=t_0 \leq t_1 \leq ... \leq t_{k^*-1} \leq t_{k^*}=1} \sum_{i=1}^{k^*} [\tilde{B}^i(t_i) - \tilde{B}^i(t_{i-1})],$$

where the k-dimensional Brownian motion $(\tilde{B}^1(t), ..., \tilde{B}^{k^*}(t))$ has the covariance matrix

$$(\Sigma)_{k^* \times k^*} = t \begin{pmatrix} 1 & \rho & \rho & ... & \rho \\ \rho & 1 & \rho & \dots & \vdots \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ \vdots & \cdots & \rho & 1 & \rho \\ \rho & \cdots & \cdots & \rho & 1 \end{pmatrix}, \tag{25}$$

where $\rho = -\dfrac{p_{max}}{1-p_{max}}$.

We now express the limiting distribution in Corollary 2.2.4 as a functional of standard Brownian motion.

Corollary 2.2.5. *Let $p_{max} = p_{j_1} = p_{j_2} = ... = p_{j_{k^*}}$, for $1 \leq j_1 < j_2 < ... < j_{k^*} \leq m$ for some $1 \leq k^* \leq m$, and let $p_i < p_{max}$, otherwise. Then*

$$\frac{LI_n - p_{max}n}{\sqrt{n}} \Longrightarrow \sqrt{p_{max}} \left\{ \frac{\sqrt{1-k^*p_{max}} - 1}{k^*} \sum_{j=1}^{k^*} B^j(1) + \max_{0=t_0 \leq t_1 \leq ... \leq t_{k^*-1} \leq t_{k^*}=1} \sum_{i=1}^{k^*} [B^i(t_i) - B^i(t_{i-1})] \right.$$

where $(B^1(t), ..., B^{k^*}(t))$ is a standard k^*-dimensional Brownian motion.

Remark 2.2.6. *: Here, for the non-uniform case, each Brownian motion $\tilde{B}^i(t)$ in Corollary 2.2.4 can be expressed as a linear combination of standard Brownian motions $(B^1(t), ..., B^k(t))$ as follows:*

$$\tilde{B}^i(t) = \beta B^i(t) + \eta \sum_{j=1, j \neq i}^{k} B^j(t), \quad i = 1, ..., k,$$

where $\beta = \dfrac{(k^*-1)\sqrt{\lambda_1} + \sqrt{\lambda_2}}{k^*}$, $\quad \eta = \dfrac{-\sqrt{\lambda_1} + \sqrt{\lambda_2}}{k^*}$,
and λ_1 and λ_2 are eigenvalues of multiplicity k-1 and 1, respectively, of covariance

matrix (25):

$$\lambda_1 = 1 - \rho = \frac{1}{1 - p_{max}}, \quad \lambda_2 = 1 + (k^* - 1)\rho = \frac{1 - k^* p_{max}}{1 - p_{max}} (< \lambda_1).$$

We have already seen several representations for the limiting law in the uniform case. Yet one more pleasing functional for the limiting distribution of LI_n is described in the following.

Theorem 2.2.7. *Let $p_{max} = p_1 = p_2 = ... = p_m = 1/m$. Then*

$$\frac{LI_n - n/m}{\sqrt{n}} \Longrightarrow \frac{\tilde{H}_m}{\sqrt{m}},$$

where

$$\tilde{H}_m = \sqrt{\frac{m-1}{m}} \max_{0=t_0 \le t_1 \le ... \le t_{m-1} \le t_m = 1} \sum_{i=1}^{m} \left[\tilde{B}^i(t_i) - \tilde{B}^i(t_{i-1}) \right] \tag{26}$$

and where $(\tilde{B}^1(t), \tilde{B}^2(t), ..., \tilde{B}^m(t))$ is an m-dimensional Brownian motion having covariance matrix (25), with $\rho = -1/(m-1)$, and thus such that $\sum_{i=1}^m \tilde{B}^i(t) = 0$, for all $0 \le t \le 1$.

2.2.4 More related expressions for the limiting distribution

Let $H_m = D_m - Z_m$, where

$$D_m = \max_{0=t_0 \le t_1 \le ... \le t_{m-1} \le t_m = 1} \sum_{i=1}^{m} [B^i(t_i) - B^i(t_{i-1})]$$

and $Z_m = \frac{1}{m} \sum_{i=1}^m B^i(1)$.

Corollary 2.2.8. *For each $m \ge 1, \tilde{H}_m =^d D_m - Z_m$, where $=^d$ denotes equality in distribution.*

Proof. Proposition 2.2.1 asserts that

$$\frac{LI_n - n/m}{\sqrt{n}} \Longrightarrow \frac{H_m}{\sqrt{m}},$$

as $n \to \infty$, while by Theorem 2.2.7,

$$\frac{LI_n - n/m}{\sqrt{n}} \Longrightarrow \frac{\tilde{H}_m}{\sqrt{m}},$$

as $n \to \infty$ as well. The conclusion follows from the definitions stated above. □

The relationship between $\tilde{H}_m$ (resp., H_m) and D_m allows us to further express the limiting distribution in a rather compact form.

Proposition 2.2.9. *Let $p_{max} = p_{j_1} = p_{j_2} = ... = p_{j_{k^*}}$, for $1 \leq j_1 < j_2 < ... < j_{k^*} \leq m$, for some $1 \leq k^* \leq m$, and let $p_i < p_{max}$, otherwise. Then*

$$\frac{LI_n - p_{max}n}{\sqrt{np_{max}}} \Longrightarrow \{\sqrt{1 - k^*p_{max}}Z_{k^*} + H_{k^*}\}.$$

For $k^* = m$, i.e., $p_{max} = 1/m$ then the result is nothing but Proposition 2.1:

$$\frac{LI_n - n/m}{\sqrt{n/m}} \Longrightarrow H_m.$$

Proof. For $k^* = m$, we have $p_{max} = 1/m$ and the result follows from Proposition 2.2.1.

For $1 \leq k^* \leq m-1$, using Corollary 2.2.5 and the definition of D_{k^*}, Z_{k^*}, and H_{k^*} together with the relation among D_{k^*}, Z_{k^*} and H_{k^*}, we get our result.

□

Remark 2.2.10. *We can also rewrite Proposition 2.2.9 as*

$$\frac{LI_n - p_{max}n}{\sqrt{p_{max}n}} \Longrightarrow (\sqrt{1 - k^*p_{max}} - 1)Z_{k^*} + D_{k^*}.$$

2.2.5 The GUE and traceless GUE

Definition 2.2.11. *Let X be an element of the $k \times k$ GUE matrix, then*

- *$X_{i,i}$ are i.i.d $N(0,1)$ for $1 \leq i \leq k$.*
- *$\overline{X_{i,j}} = X_{j,i}; Re(X_{i,j}), Im(X_{i,j})$ are iid $N(0, 1/2)$ for $1 \leq i < j \leq k$.*
- *$X_{i,i}, Re(X_{i,j}), Im(X_{i,j})$ are mutually independent.*

Definition 2.2.12. *An element of the $k \times k$ traceless GUE is of form $X - tr(X)/k * I_{k\times k}$, where X is an element of the $k \times k$ GUE, whose trace is denoted by $tr(X)$.*

There is an important relationship between GUE and GUE traceless

Theorem 2.2.13. *For $k \geq 2$, let X be an $k \times k$ GUE matrix. Let $X^0 = X - tr(X)/k * I_{k\times k}$ be the corresponding traceless GUE. Denote $\lambda_{1,k}$ and $\lambda^0_{1,k}$ be the maximum eigenvalue of X and X^0, respectively. Then*

$$\lambda_{1,k} =^d \lambda^0_{1,k} + Z_k,$$

where $Z_k \sim N(0, 1/k)$. Moreover, λ^0_1 and Z_k are independent.

An important result obtained for the largest eigenvalue of an $k \times k$ GUE matrix is Tracy-Widom (1994) [35]

Theorem 2.2.14.

$$k^{1/6}(\lambda_{1,k} - 2\sqrt{k}) \Longrightarrow F_2$$

where $\lambda_{1,k}$ is the largest eigenvalue of an $k \times k$ GUE matrix, and F_2 is the Tracy-Widom distribution, i.e., the c.d.f of Tracy-Widom distribution is:

$$F_2(t) = \exp(-\int_t^\infty (x-t)^2 u^2(x)dx), \quad t \in \mathbb{R}$$

where $u(x)$ is the solution to the Painlevé II equation $u_{xx} = 2u^3 + xu$ with $u \sim Ai(x)$, as $x \to \infty$, where $Ai(x)$ is the Airy function given by

$$Ai(x) = \frac{1}{2\pi}\int_{-\infty}^{\infty} \exp\left[i\left(xt + \frac{t^3}{3}\right)\right] dt$$

Also, it is well known that

$$\frac{\lambda_{1,k}}{\sqrt{k}} \to 2$$

a.s. and in L^1.

Another important result that links everything together due to Baryshnikov (2001) [3]

Theorem 2.2.15. *The process $(D_m)_{m\geq 1}$, as defined above, is identical in law to $(\lambda_{1,m})_{m\geq 1}$, the process consisting of the largest eigenvalues of the first m rows and m columns of an infinite GUE matrix.*

Hence,

$$(D_m - 2\sqrt{m})m^{1/6} \Longrightarrow F_2, \text{ as } m \to \infty$$

From these results, the asymtotics of H_m follows:

Theorem 2.2.16.

$$\frac{H_m}{\sqrt{m}} \to 2$$

a.s. and in L^1, as $m \to \infty$. Moreover,

$$\left(\frac{H_m}{\sqrt{m}} - 2\right) m^{2/3} \Longrightarrow F_2,$$

where F_2 is the Tracy-Widom distribution. The same statements hold for $\tilde{H}_m$ in place of H_m.

For the uniform case, from Proposition 2.2.9, we have

$$\frac{LI_n - n/m}{\sqrt{n/m}} \Longrightarrow H_m,$$

which is (in law) the maximum eigenvalue of a $m \times m$ element of the traceless GUE.

2.3 Simulation study of LI_n and max eigenvalue of GUE

2.3.1 Idea of simulation

Recall Proposition 2.2.9,

$$\frac{LI_n - p_{max}n}{\sqrt{np_{max}}} \Rightarrow \sqrt{1 - k^* p_{max}} Z_{k^*} + H_{k^*}.$$

We shall use simulation to generate the samples of the LI_n and H_{k^*} and D_{k^*} and then draw the histogram to see their shape:

- First we must fix values of $p_1, p_2, ..., p_m \in (0,1)$. Let k^* be number multiplicity of p_{max}
- We generate the sequence $(R_i)_{1\leq i\leq n}$ i.i.d with

$$\mathbb{P}(R_1 = \alpha_r) = p_r, 1 \leq r \leq m. \tag{27}$$

- For simplicity, we can consider $\alpha_r = r, 1 \leq r \leq m$.
- For the right hand side, we have $Z_{k^*} = \frac{1}{k^*}\sum_{i=1}^{k^*} B^i(1)$, where $(B^1(t), ..., B^m(t))$ is a standard m-dimensional Brownian motion. Hence $Z_{k^*} \sim N(0, 1/k^*)$.
- Note that $Z_{k^*} \Longrightarrow 0$ as $k^* \to \infty$
- H_{k^*} is (in law) the maximum eigenvalue of a $k^* \times k^*$ element of traceless GUE.
- D_{k^*} is (in law) the maximum eigenvalue of a $k^* \times k^*$ element of GUE.

2.3.2 Procedure of generating LI_n

First, we generate $R_1, ..., R_n$ satisfying (27) by the following manner: For each i from 1 to n,

- Generate u from $U(0,1)$.

  If $u < p_1$, let $W_i = 1$.

  If $p_1 \leq u < p_1 + p_2$, let $R_i = 2$.

  If $p_1 + p_2 \leq u < p_1 + p_2 + p_3$, let $R_i = 3$.

  .....

  If $p_1 + p_2 + ... + p_{m-1} \leq u < 1$, let $R_i = m$.

- For the sequence above, we can compute LI_n

Note that for the uniform case, we can just generate $R_i, 1 \leq i \leq n$ from Uniform discrete distribution $U(m)$, and here $k^* = m$.

2.3.3 Procedure of generating an element of the GUE and of the traceless GUE

We generate a $k \times k$ GUE matrix $\mathbf{X}$ by

$$\begin{pmatrix} X_1 & Y_1 + iZ_1 & Y_2 + iZ_2 & ... & Y_{k^*-1} + iZ_{k^*-1} \\ Y_1 - iZ_1 & X_2 & Y_k + iZ_k & \dots & \vdots \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ \vdots & \cdots & \cdots & X_{k^*-1} & Y_{(k^*-1)k^*/2} + iZ_{(k^*-1)k^*/2} \\ Y_{k^*-1} - iZ_{k^*-1} & \cdots & \cdots & Y_{(k^*-1)k^*/2} - iZ_{(k^*-1)k^*/2} & X_{k^*} \end{pmatrix},$$

where X_i, Y_i, Z_i are respectively $N(0,1), N(0,1/2), N(0,1/2)$, and are mutually independent.

Now define $\mathbf{X}^0 = \mathbf{X} - tr(\mathbf{X})/k * I_{k^* \times k^*}$, and let $D_{k^*} =$ max eigenvalue of $\mathbf{X}$ and $H_{k^*} =$ max eigenvalue of $\mathbf{X}^0$.

2.3.4 Histograms of LI_n, H_m, and D_m

Using the procedures above, we will simulate the histogram of $\frac{LI_n - np_{max}}{\sqrt{np_{max}}}$ for both uniform and non-uniform cases, H_m and D_m for $n = 10^3, m = 50$. In addition, we denote *numRep* as the number of sample replications. Throughout, we let $numRep = 10^4$. We also draw the QQ-plots to verify that these random variables are not normally distributed.

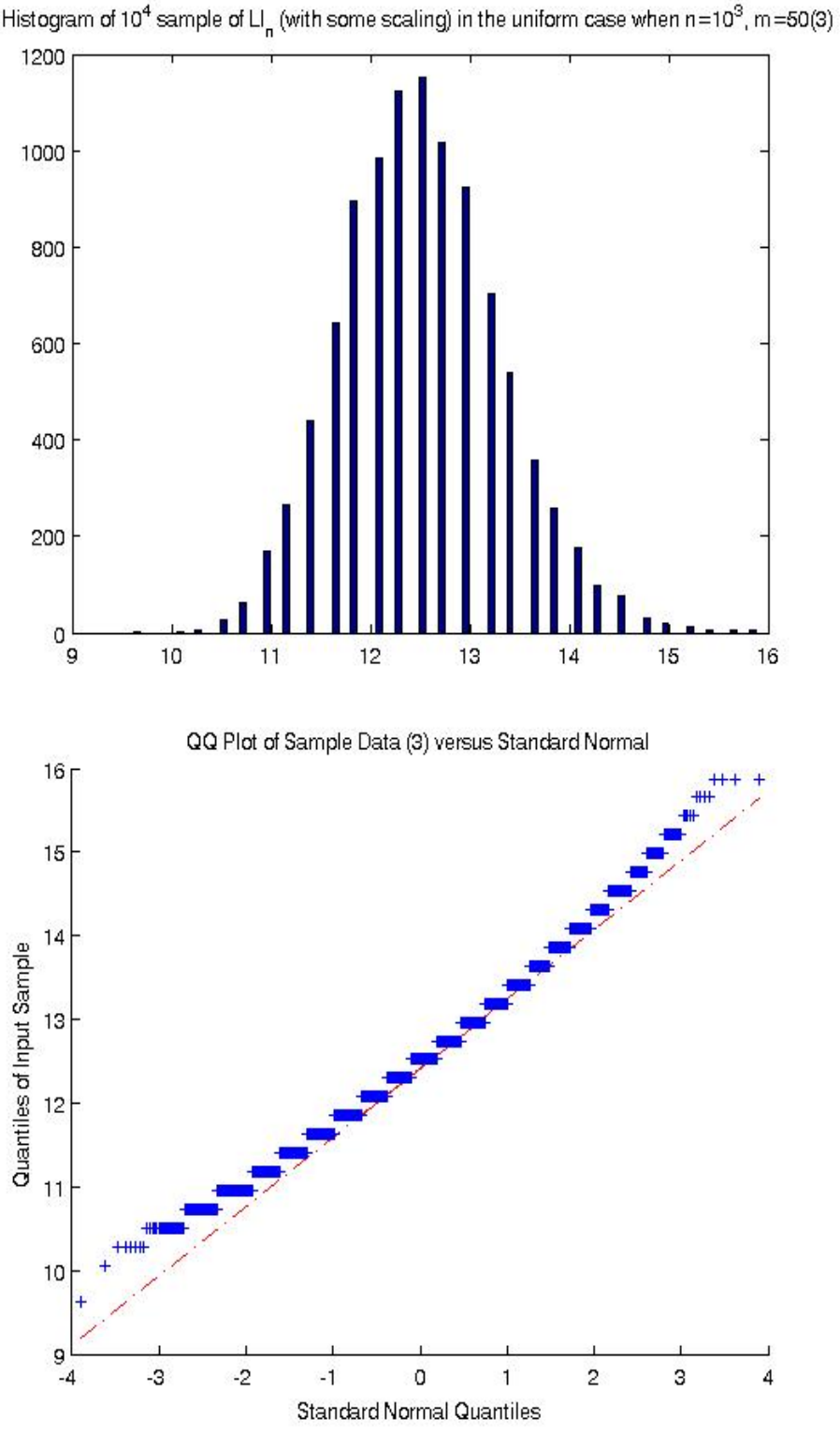


Figure 1: Histogram and QQ-plot of LI_n in the uniform case where $numRep = 10^4, n = 10^3, m = 50$.

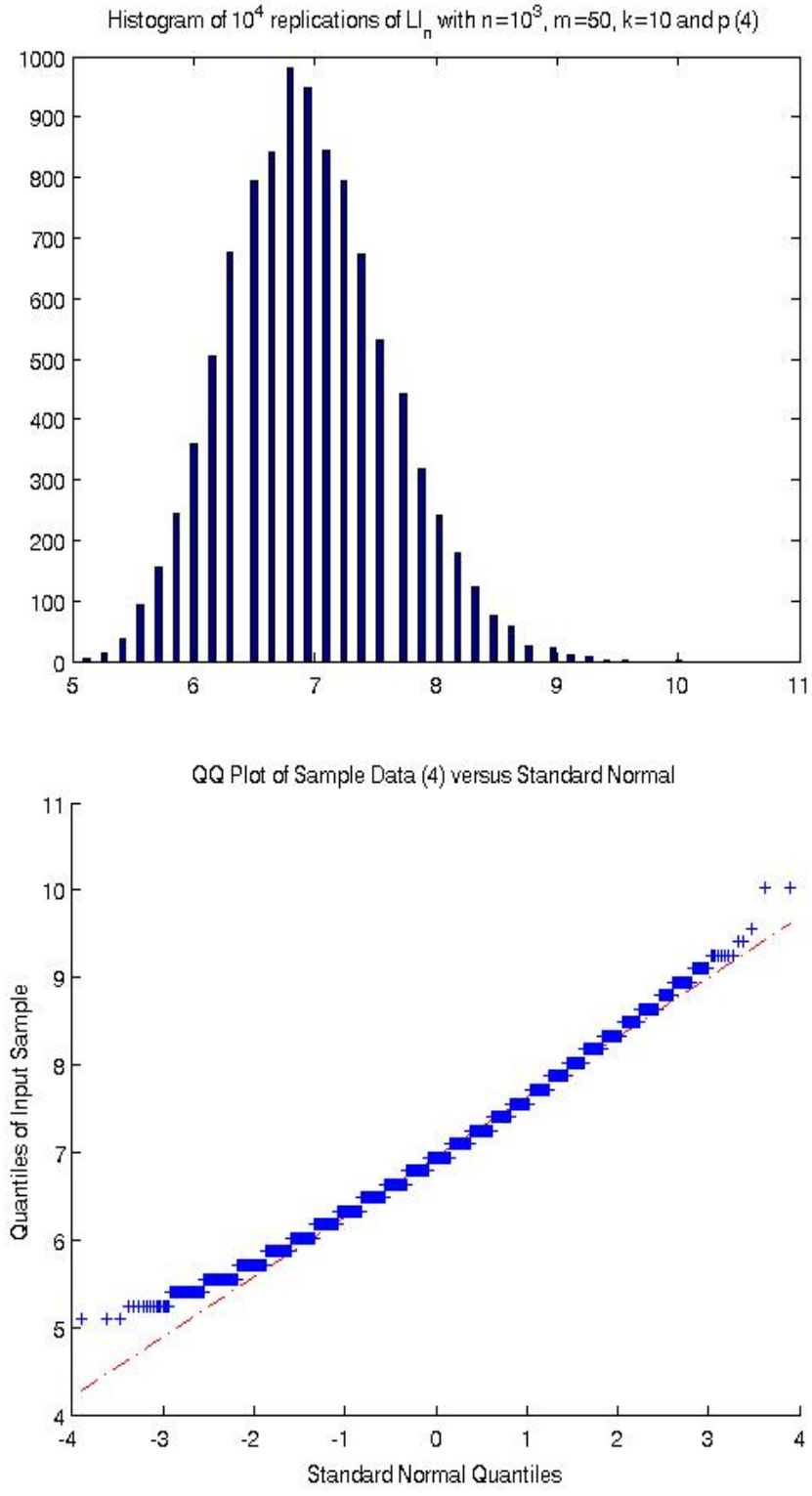


Figure 2: Histogram and QQ-plot of LI_n in the non-uniform case where $numRep = 10^4, n = 10^3, m = 50, k = 10$.

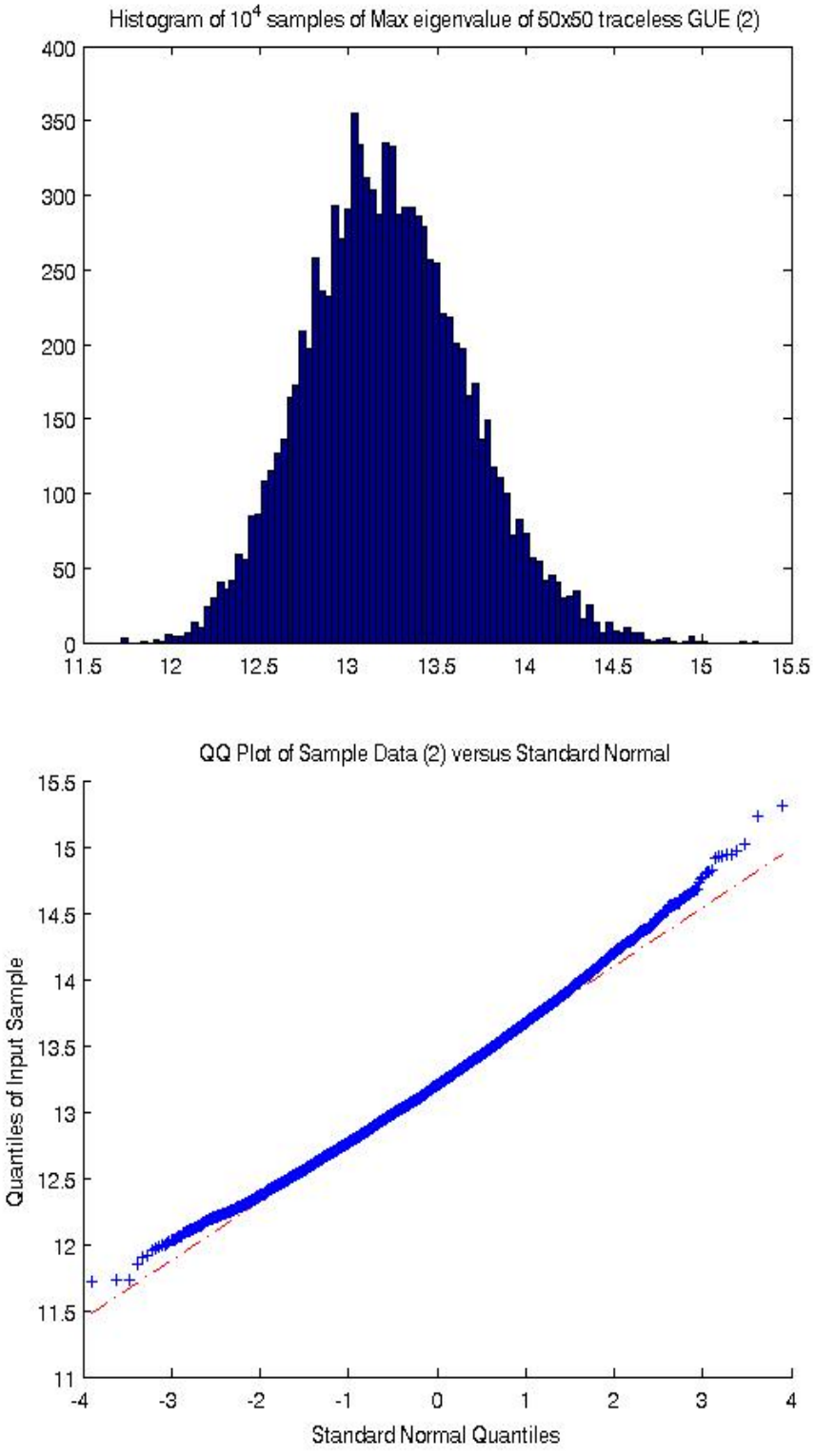


Figure 3: Histogram and QQ-plot of H_m in the uniform case where $numRep = 10^4, m = 50$.

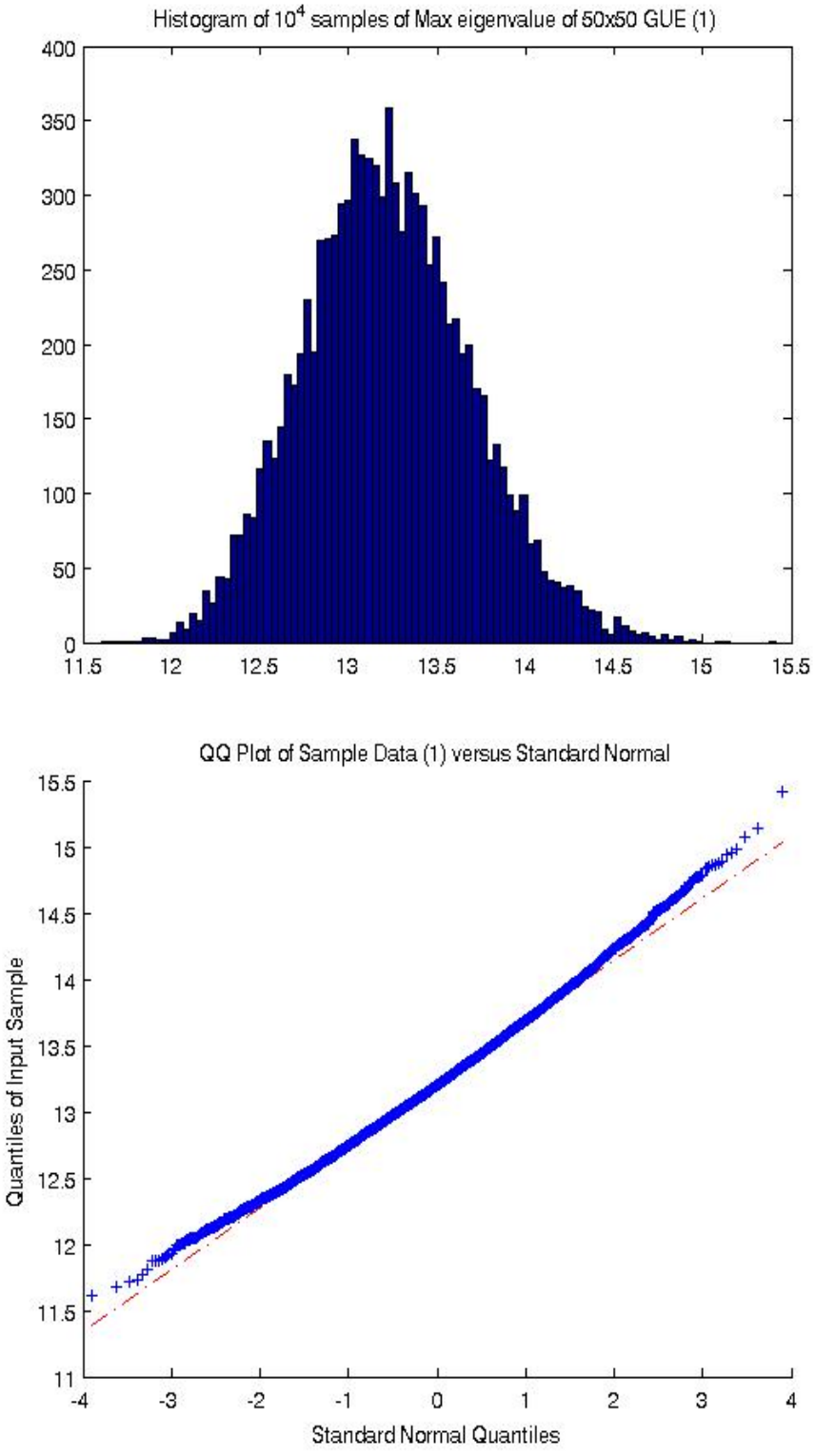


Figure 4: Histogram and QQ-plot of D_m in the non-uniform case where $numRep = 10^4, m = 50$.

2.3.5 Comparing LI_n with H_m

Now we verify that $\dfrac{LI_n - n/m}{\sqrt{n/m}}$ where LI_n is simulated in the uniform case and H_m are identical in distribution. To illustrate, we set $m = 50$ and plot the historgrams of $\dfrac{LI_n - n/m}{\sqrt{n/m}}$ for $n = 10^3, 10^4, 2 * 10^4$. The fourth figure is the histogram of H_{50}, the maximum eigenvalue of an element of 50×50 traceless GUE.

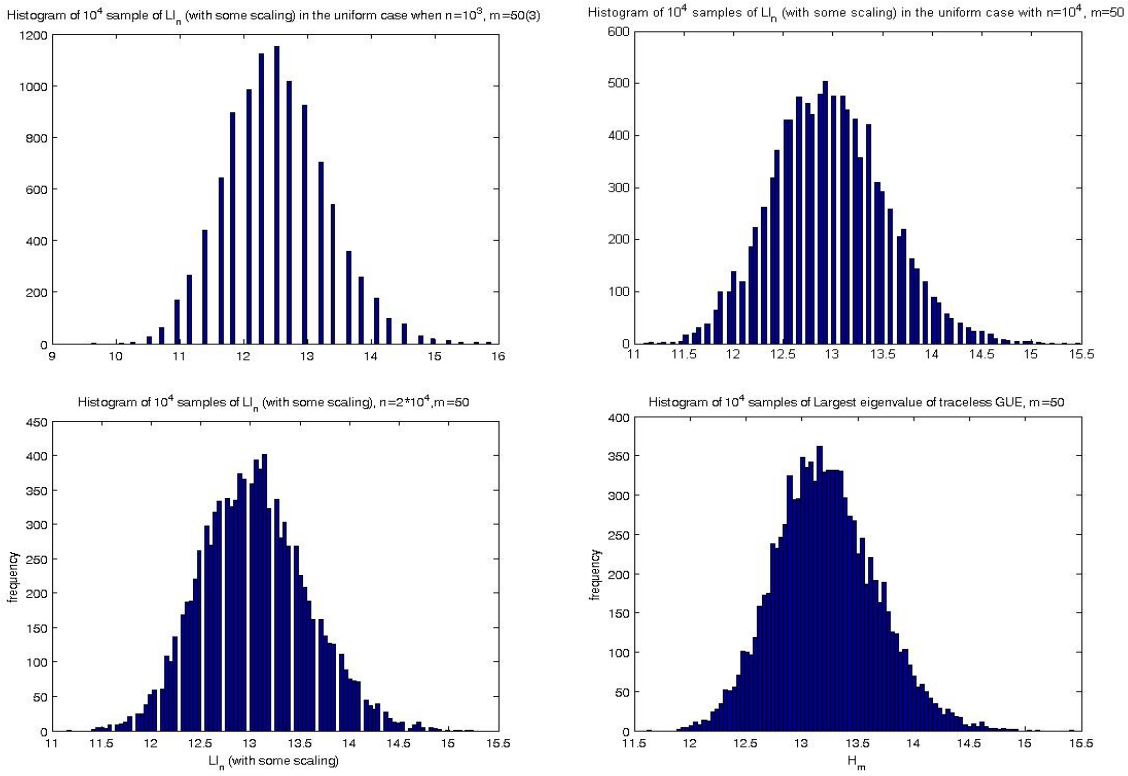


Figure 5: Histogram of LI_n for different values of n. From left to right, top to bottom: $n = 1000$, $n = 10000$, $n = 20000$ and $n \to \infty$.

To illustrate further the relationship between LI_n and H_m, we perform a scatter plot and QQ-plot for the simulated values of $\dfrac{LI_n - n/m}{\sqrt{n/m}}$ and H_m for $m = 50, n = 2 * 10^4$.

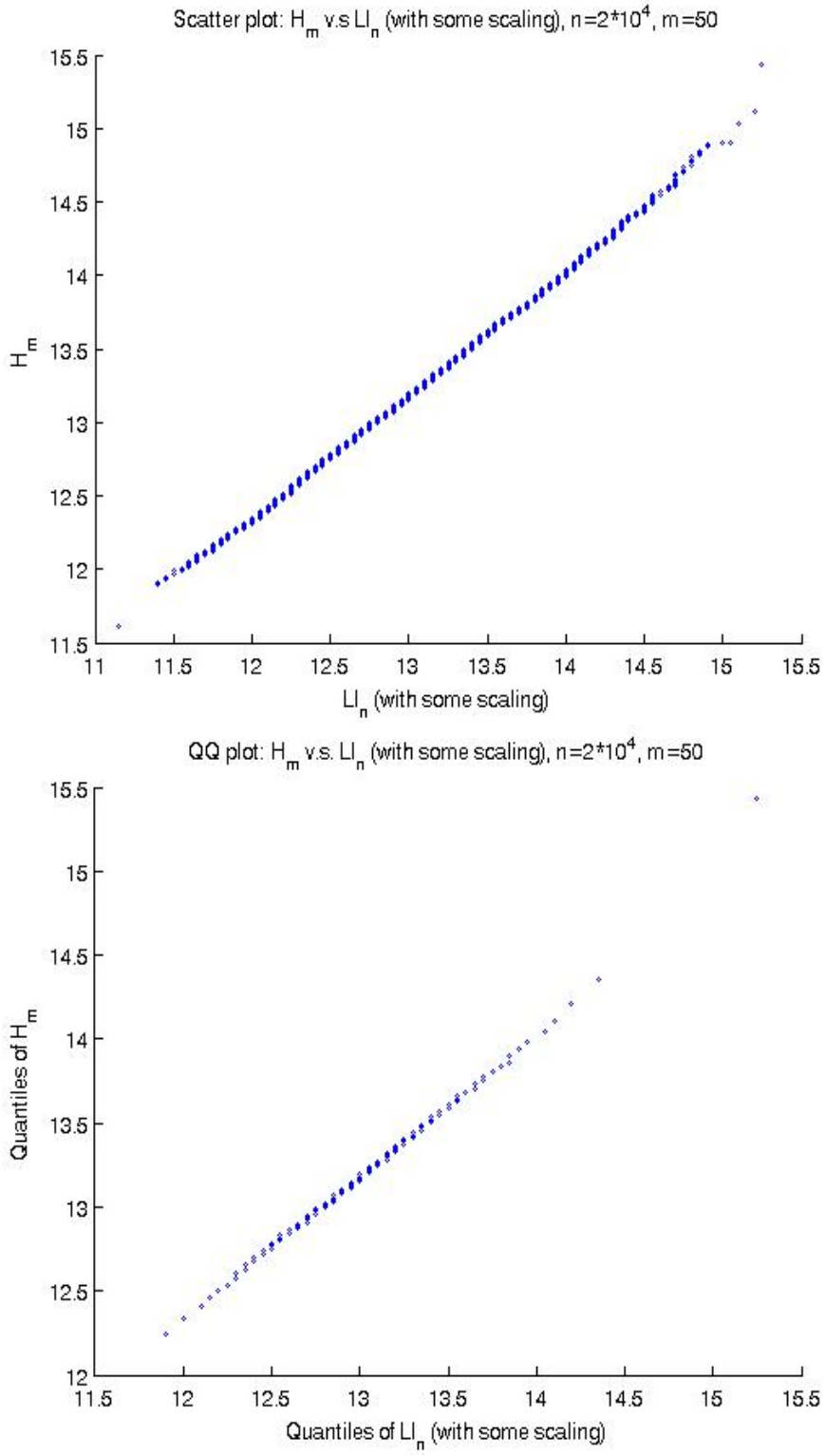


Figure 6: Scatter plot and QQ-plot of H_m vs LI_n in the uniform case where $numRep = 10^4, n = 2 * 10^4, m = 50$.

2.4 Increasing number of classes, i.e., $m = m(n) \to \infty$

We now consider the case where the size m of the alphabet varies with n and further we assume the multiplicity $k^*(n)$ of p_{max} goes to infinity as $n \to \infty$. The limiting distribution of LI_n is studied via the first row of the Young tableau in Theorem 6 in [6]. We can rewrite the Theorem 6 in term of LI_n as the following

Theorem 2.4.1. *Let $p_{max} = \max_{j=1,\dots,m} p_j$. Suppose $k^* = k^*(n) \to \infty$ as $n \to \infty$ in such way that*

$$\frac{(k^*)^{7/10}}{p_{max}^{3/10}} = o\left(n^{3/10}(\log n)^{-3/5}\right).$$

Assume, moreover, that

$$p_{2nd}^2 \frac{n^{11/10}}{(\log n)^{1/5}} = o(p_{max}),$$

where $p_{2nd} = max\{p_j < p_{max}, j = 1, ..., m\}$. Then as $n \to \infty$,

$$\left(\frac{LI_n - np_{max}}{\sqrt{np_{max}}} - 2\sqrt{k^*}\right)(k^*)^{1/6} \Longrightarrow F_2,$$

where F_2 denotes the Tracy-Widom distribution.

For the uniform case, the result follows Theorem 4 and Corrolary 5 in [6]:

Theorem 2.4.2. *Let m tend to infinity as $n \to \infty$ in such way that $m = o\left(n^{3/10}(\log n)^{-3/5}\right)$. Then as $n \to \infty$,*

$$\left(\frac{LI_n - n/m}{\sqrt{n/m}} - 2\sqrt{m}\right) m^{1/6} \Longrightarrow F_2.$$

CHAPTER III

COMPARISON OF THE TWO ESTIMATORS

3.1 Results

As we have seen, Chapters 1 and 2 introduce two estimators for p_{max} and their limiting distribution. In this chapter, we see how good the estimators are by comparing at their mean square error and bias corrected estimator:

Proposition 3.1.1. *In comparison of the mean square error,* $\frac{X_{(m)}}{n}$ *performs better in estimating* p_{max} *than* $\frac{LI_n}{n}$.

Proposition 3.1.2. *In comparison the bias corrected estimator,* $\frac{LI_n}{n}$ *performs better in estimating* p_{max} *than* $\frac{X_{(m)}}{n}$.

The detail of the comparisons is discuss in the following sections.

3.2 $\frac{X_{(m)}}{n}$ *versus* $\frac{LI_n}{n}$

Recall that in Theorem 1.3.2 and Theorem 2.4.1, we obtained the limiting distribution of $X_{(m)}$ and LI_n with natural standardization:

$$a_{k^*}\left(\frac{X_{(m)}-np_{max}}{\sqrt{np_{max}}}-b_{k^*}\right)\Longrightarrow G, \tag{28}$$

where $a_{k^*}=(2\log k^*)^{1/2}$ while $b_{k^*}=(2\log k^*)^{1/2}-\frac{1}{2}(2\log k^*)^{-1/2}\left(\log\log k^*+\log 4\pi\right)$, and

$$(k^*)^{1/6}\left(\frac{LI_n-np_{max}}{\sqrt{np_{max}}}-2\sqrt{k^*}\right)\Longrightarrow F_2. \tag{29}$$

Next,

$$MSE(X_{(m)}/n)=bias^2(X_{(m)}/n)+Var(X_{(m)}/n),\quad bias(X_{(m)}/n)=\mathbb{E}(X_{(m)}/n)-p_{max},$$

and similarly,

$$MSE(LI_n/n) = bias^2(LI_n/n) + Var(LI_n/n), \quad bias(LI_n/n) = \mathbb{E}(LI_n/n) - p_{max}.$$

Hence using the limiting distribution in (28) and (29), we obtain the asymptotic of the estimators' bias and variance as $n \to \infty$:

$$bias\left(\frac{X_{(m)}}{n}\right) \sim \sqrt{\frac{p_{max}}{n}}\left(\frac{\mathbb{E}(G)}{a_{k^*}} + b_{k^*}\right), \quad Var\left(\frac{X_{(m)}}{n}\right) \sim \frac{p_{max}Var(G)}{na_{k^*}^2},$$

and similarly,

$$bias\left(\frac{LI_n}{n}\right) \sim \sqrt{\frac{p_{max}}{n}}\left(\frac{\mathbb{E}(F_2)}{(k^*)^{1/6}} + 2\sqrt{k^*}\right), \quad Var\left(\frac{LI_n}{n}\right) \sim \frac{p_{max}Var(F_2)}{n(k^*)^{1/3}}.$$

Thus,

$$MSE\left(\frac{X_{(m)}}{n}\right) \sim \frac{p_{max}}{n}\left(\frac{\mathbb{E}(G)}{a_{k^*}} + b_{k^*}\right)^2 + \frac{p_{max}Var(G)}{na_{k^*}^2} \sim \frac{p_{max}b_{k^*}^2}{n}, \tag{30}$$

and

$$MSE\left(\frac{LI_n}{n}\right) \sim \frac{p_{max}}{n}\left(\frac{\mathbb{E}(F_2)}{(k^*)^{1/6}} + 2\sqrt{k^*}\right)^2 + \frac{p_{max}Var(F_2)}{n(k^*)^{1/3}} \sim \frac{4p_{max}k^*}{n}. \tag{31}$$

Therefore, as $n \to \infty$ and $k^* = k^*(n) \to \infty$,

$$MSE\left(\frac{X_{(m)}}{n}\right) << MSE\left(\frac{LI_n}{n}\right),$$

and $\dfrac{X_{(m)}}{n}$ is "a better estimator" for p_{max}.

3.3 Bias Corrected Estimators

The results in (28) and (29) can be rewritten as

$$\frac{a_{k^*}\sqrt{n}}{\sqrt{p_{max}}}\left(\frac{X_{(m)}}{n} - \frac{b_{k^*}\sqrt{p_{max}}}{\sqrt{n}} - p_{max}\right) \Longrightarrow G, \tag{32}$$

$$\frac{(k^*)^{1/6}\sqrt{n}}{\sqrt{p_{max}}}\left(\frac{LI_n}{n} - \frac{2\sqrt{k^*p_{max}}}{\sqrt{n}} - p_{max}\right) \Longrightarrow F_2. \tag{33}$$

Let

$$\hat{p}^1_{max} = \frac{X_{(m)}}{n} - \frac{b_{k^*}\sqrt{\frac{X_{(m)}}{n}}}{\sqrt{n}} - \frac{\mathbb{E}(G)\sqrt{\frac{X_{(m)}}{n}}}{a_{k^*}\sqrt{n}}$$

and

$$\hat{p}^2_{max} = \frac{LI_n}{n} - \frac{2(k^*)^{1/2}\sqrt{\frac{LI_n}{n}}}{\sqrt{n}} - \frac{\mathbb{E}(F_2)\sqrt{\frac{LI_n}{n}}}{(k^*)^{1/6}\sqrt{n}}.$$

Then, as $n \to \infty$,

$$bias(\hat{p}^1_{max}) = \mathbb{E}\left(\frac{X_{(m)}}{n}\right) - p_{max} - \frac{b_{k^*}\mathbb{E}(\sqrt{X_{(m)}/n})}{\sqrt{n}} - \frac{\mathbb{E}(G)\mathbb{E}(\sqrt{X_{(m)}/n})}{a_{k^*}\sqrt{n}}$$

$$\sim \frac{\mathbb{E}(G)\sqrt{p_{max}}}{a_{k^*}\sqrt{n}} + \frac{b_{k^*}\sqrt{p_{max}}}{\sqrt{n}} - \frac{b_{k^*}\mathbb{E}(\sqrt{X_{(m)}/n})}{\sqrt{n}} - \frac{\mathbb{E}(G)\mathbb{E}(\sqrt{X_{(m)}/n})}{a_{k^*}\sqrt{n}}$$

$$= \frac{1}{\sqrt{n}}\left(\frac{\mathbb{E}(G)}{a_{k^*}} + b_{k^*}\right)\left(\sqrt{p_{max}} - \mathbb{E}\left(\sqrt{\frac{X_{(m)}}{n}}\right)\right).$$

Now, for the variance,

$$Var(\hat{p}^1_{max}) = Var\left(\frac{X_{(m)}}{n} - \frac{1}{\sqrt{n}}(b_{k^*} + \frac{\mathbb{E}(G)}{a_{k^*}})\sqrt{\frac{X_{(m)}}{n}}\right)$$

$$\sim \frac{Var(G)p_{max}}{a^2_{k^*}n} + \frac{1}{n}\left(b_{k^*} + \frac{\mathbb{E}(G)}{a_{k^*}}\right)^2\left(\sqrt{p_{max}} - \mathbb{E}(\sqrt{X_{(m)}/n})\right)\left(\sqrt{p_{max}} + \mathbb{E}(\sqrt{X_{(m)}/n})\right)$$
$$+\frac{\sqrt{p_{max}}}{n^{3/2}}\left(b_{k^*} + \frac{\mathbb{E}(G)}{a_{k^*}}\right)^3 - \frac{2}{n^2}\left(b_{k^*} + \frac{\mathbb{E}(G)}{a_{k^*}}\right)Cov\left(X_{(m)}, \sqrt{X_{(m)}}\right).$$

Hence,

$$MSE(\hat{p}^1_{max}) = bias^2(\hat{p}^1_{max}) + Var(\hat{p}^1_{max})$$

$$= \frac{Var(G)p_{max}}{a^2_{k^*}n} + \frac{2\sqrt{p_{max}}}{n}\left(b_{k^*} + \frac{\mathbb{E}(G)}{a_{k^*}}\right)^2\left(\sqrt{p_{max}} - \mathbb{E}(\sqrt{X_{(m)}/n})\right)$$
$$+\frac{\sqrt{p_{max}}}{n^{3/2}}\left(b_{k^*} + \frac{\mathbb{E}(G)}{a_{k^*}}\right)^3 - \frac{2}{n^2}\left(b_{k^*} + \frac{\mathbb{E}(G)}{a_{k^*}}\right)Cov\left(X_{(m)}, \sqrt{X_{(m)}}\right).$$

From (32) and the fact that $p_{max}, \frac{b_{k^*}\sqrt{p_{max}}}{\sqrt{n}}, \frac{\sqrt{p_{max}}}{a_{k^*}\sqrt{n}} \to 0$ and $b_{k^*}\sqrt{p_{max}} \to 0$ as $n \to \infty$, we get

$$\mathbb{E}\left(\sqrt{\frac{X_{(m)}}{n}}\right) - \sqrt{p_{max}} \sim \frac{1}{n^{1/4}}. \tag{34}$$

This implies

$$MSE(\hat{p}^1_{max}) \sim \frac{p_{max}}{a^2_{k^*} n}. \tag{35}$$

Let us now consider

$$\hat{p}^2_{max} = \frac{LI_n}{n} - \frac{2(k^*)^{1/2}\sqrt{\frac{LI_n}{n}}}{\sqrt{n}} - \frac{\mathbb{E}[F_2]\sqrt{\frac{LI_n}{n}}}{(k^*)^{1/6}\sqrt{n}}.$$

Then we have

$$bias(\hat{p}^2_{max}) = \mathbb{E}\left(\frac{LI_n}{n}\right) - \frac{2\sqrt{k^*}}{\sqrt{n}}\mathbb{E}\left(\sqrt{\frac{LI_n}{n}}\right) - \frac{\mathbb{E}(F_2)}{(k^*)^{1/6}\sqrt{n}}\mathbb{E}\left(\sqrt{\frac{LI_n}{n}}\right)$$

$$\sim \frac{1}{\sqrt{n}}\left(2\sqrt{k} + \frac{\mathbb{E}(F_2)}{(k^*)^{1/6}}\right)\left[\sqrt{p_{max}} - \mathbb{E}\left(\sqrt{\frac{LI_n}{n}}\right)\right] \text{ from (33).}$$

Furthermore,

$$Var(\hat{p}^2_{max}) = Var\left(\frac{LI_n}{n} - \frac{1}{\sqrt{n}}(\sqrt{k^*} + \frac{\mathbb{E}(F_2)}{(k^*)^{1/6}})\sqrt{\frac{X_{(m)}}{n}}\right)$$

$$\sim \frac{Var(F_2)p_{max}}{(k^*)^{1/3}n} + \frac{1}{n}\left(\sqrt{k^*} + \frac{\mathbb{E}(F_2)}{(k^*)^{1/6}}\right)^2\left(\sqrt{p_{max}} - \mathbb{E}(\sqrt{LI_n/n})\right)\left(\sqrt{p_{max}} + \mathbb{E}(\sqrt{LI_n/n})\right)$$
$$+\frac{\sqrt{p_{max}}}{n^{3/2}}\left(\sqrt{k^*} + \frac{\mathbb{E}(F_2)}{(k^*)^{1/6}}\right)^3 - \frac{2}{n^2}\left(\sqrt{k^*} + \frac{\mathbb{E}(F_2)}{(k^*)^{1/6}}\right)Cov\left(LI_n, \sqrt{LI_n}\right).$$

Thus,

$$MSE(\hat{p}^2_{max}) \sim \frac{Var(F_2)p_{maX}}{(k^*)^{1/3}n} + \frac{2\sqrt{p_{max}}}{n}\left(\sqrt{k^*} + \frac{\mathbb{E}(F_2)}{(k^*)^{1/6}}\right)^2\left(\sqrt{p_{max}} - \mathbb{E}(\sqrt{LI_n/n})\right)$$
$$+\frac{\sqrt{p_{max}}}{n^{3/2}}\left(\sqrt{k^*} + \frac{\mathbb{E}(F_2)}{(k^*)^{1/6}}\right)^3 - \frac{2}{n^2}\left(\sqrt{k^*} + \frac{\mathbb{E}(F_2)}{(k^*)^{1/6}}\right)Cov\left(LI_n, \sqrt{LI_n}\right).$$

From [14] we obtain

$$\mathbb{E}\left(\sqrt{\frac{LI_n}{n}}\right) - \sqrt{p_{max}} \sim \frac{1}{n^{1/2}}. \tag{36}$$

Then it can be shown (especially in the uniform case) that

$$MSE(\hat{p}^2_{max}) \sim \frac{p_{max}}{(k^*)^{1/3}n}. \tag{37}$$

It is clear that $\frac{p_{max}}{(k^*)^{1/3}n} << \frac{p_{max}}{a^2_{k^*}n}$. Comparing (35) and (37) asymptotically, we obtain that

$$MSE(\hat{p}^2_{max}) << MSE(\hat{p}^1_{max}).$$

Hence, under the derivations (34) and (36) we conclude that $\hat{p}^2_{max}$ performs better in estimating p_{max}.

CHAPTER IV

CONSTRUCTING CONFIDENCE INTERVALS FOR THE MAXIMUM PROBABILITY

4.1 Motivation

Though out the thesis, we see a strong connection between p_{max} and its multiplicity k^*. If $k^* = 1$, the limiting distribution of the multinomial max with standardization is a standard normal distribution. In case k^* approaches infinity, the limiting distribution is Gumbel. Assumptions (8) and (9) provide some relations of the two quantities p_{max} and k^*. Also, in constructing the confidence interval for p_{max}, the two quantities have a very tight relation. In deed, from (28), we can obtain the confidence interval of p_{max} using Gumbel distribution G. Let $g_{1-\alpha/2}$ and $g_{\alpha/2}$ be the $(1-\alpha/2)*100th$ and $(\alpha/2)*100th$ percentile of G. Then we have $g_{1-\alpha/2} = -\log\log\dfrac{2}{2-\alpha}$ and $g_{\alpha/2} = -\log\log\dfrac{2}{\alpha}$. Via (28), the $(1-\alpha/2)*100\%$ CI for p_{max} can be constructed as

$$-\log\log\frac{2}{\alpha} \le a_{k^*}\left(\frac{X_{(m)} - np_{max}}{\sqrt{np_{max}}} - b_{k^*}\right) \le -\log\log\frac{2}{2-\alpha}.$$

Replacing p_{max} by $\dfrac{X_{(m)}}{n}$ in the denominator and solving for p_{max} in the numerator, again, we obtain

$$\frac{X_{(m)}}{n} + \left(\frac{\log\log\frac{2}{2-\alpha}}{a_{k^*}} + b_{k^*}\right)\frac{\sqrt{X_{(m)}}}{n} \le p_{max} \le \frac{X_{(m)}}{n} + \left(\frac{\log\log\frac{2}{\alpha}}{a_{k^*}} + b_{k^*}\right)\frac{\sqrt{X_{(m)}}}{n}.$$

We want to construct a 95% CI for p_{max}. Note that $g_{.975} = 3.676$ and $g_{.025} = -1.305$, then the 95% CI of p_{max} is

$$\boxed{\frac{X_{(m)}}{n} - \left(b_{k^*} + \frac{3.676}{a_{k^*}}\right)\frac{\sqrt{X_{(m)}}}{n} \le p_{max} \le \frac{X_{(m)}}{n} - \left(b_{k^*} - \frac{1.305}{a_{k^*}}\right)\frac{\sqrt{X_{(m)}}}{n}.} \quad (38)$$

Note that in order to obtain the confidence interval, we need to know k^*. In the following sections, we construct the confidence interval in the case $k^* = 1$ and

introduce two methods to construct confidence interval for p_{max} when k^* gets large: estimating k^* and refining the confidence interval.

4.2 When $k^* = 1$

Let $z_{1-\alpha/2}$ be the $(1-\alpha/2)*100\%$ percentile of $N(0,1)$. Then the $(1-\alpha/2)*100\%$ confidence interval (CI) for p_{max} can be constructed as

$$-z_{1-\alpha/2} \leq \frac{X_{(m)} - np_{max}}{\sqrt{np_{max}(1-p_{max})}} \leq z_{1-\alpha/2}.$$

Replacing p_{max} by $\frac{X_{(m)}}{n}$ in the denomiator and solving the inequalities above for p_{max} in the numerator, we obtain

$$\frac{X_{(m)}}{n} - \frac{z_{1-\alpha/2}}{n}\sqrt{X_{(m)}(1-\frac{X_{(m)}}{n})} \leq p_{max} \leq \frac{X_{(m)}}{n} + \frac{z_{1-\alpha/2}}{n}\sqrt{X_{(m)}(1-\frac{X_{(m)}}{n})}.$$

Suppose we want to construct a 95% CI of p_{max}. Note that $z_{.975} = 1.96$.Then the 95% CI of p_{max} is

$$\boxed{\frac{X_{(m)}}{n} - \frac{1.96}{n}\sqrt{X_{(m)}(1-\frac{X_{(m)}}{n})} \leq p_{max} \leq \frac{X_{(m)}}{n} + \frac{1.96}{n}\sqrt{X_{(m)}(1-\frac{X_{(m)}}{n})}.}$$

4.3 When k^* gets large: Estimating the Multiplicity of the Maximum Probability

In order to construct the confidence integral for p_{max} in (38), we need to know k^*. In this section, we develop a method that estimates k^* from a given sample. Recall $(X_1, ..., X_m)$ is a multinomial distribution with parameters $n, p_1, ..., p_m$. Recall also that $p_{max} = \max_{j=1,...,m} p_j$ with multiplicity k^*. We prove that p_{max} is in fact the largest eigenvalue of the covariance matrix of $(X_1, ..., X_m)$ with multiplicity $k^* - 1$. The covariance matrix can be estimated from a preliminary sample. We then estimate the multiplicity of the largest eigenvalue of this covariance matrix and hence produce an estimator for k^*.

4.3.1 Maximum probability p_{max} is an eigenvalue of the covariance matrix with multiplicity $k^* - 1$

Consider the covariance matrix Σ_m for $X_1, X_2, ..., X_m$:

$$\Sigma_m = \begin{pmatrix} p_1(1-p_1) & -p_1p_2 & -p_1p_3 & ... & -p_1p_m \\ -p_2p_1 & p_2(1-p_2) & -p_2p_3 & \dots & \vdots \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ \vdots & \cdots & -p_{m-1}p_{m-2} & p_{m-1}(1-p_{m-1}) & -p_{m-1}p_m \\ -p_mp_1 & \cdots & \cdots & -p_mp_{m-1} & p_m(1-p_m) \end{pmatrix}.$$

We show that p_{max} is an eigenvalue of Σ_m with multiplicity $k^* - 1$. First, consider the uniform case:

$$\Sigma_m^{uniform} = \frac{1}{m^2} \begin{pmatrix} m-1 & -1 & -1 & ... & -1 \\ -1 & m-1 & -1 & \dots & \vdots \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ \vdots & \cdots & -1 & m-1 & -1 \\ -1 & \cdots & \cdots & -1 & m-1 \end{pmatrix}.$$

We can easily verify that $\frac{1}{m}$ is an eigenvalue with multiplicity $m-1$. The following are associated eigenvectors for $\frac{1}{m}$:

$$v_1 = (1, -1, 0, 0, ..., 0, 0), v_2 = (1, 0, -1, 0, ..., 0, 0), ..., v_{m-1} = (1, 0, 0, 0, ..., 0, -1)$$

Now consider an $k^* \times k^*$ covariance matrix

$$\Sigma_{k^*,p_{max}} = \begin{pmatrix} p_{max}(1-p_{max}) & -p_{max}^2 & -p_{max}^2 & ... & -p_{max}^2 \\ -p_{max}^2 & p_{max}(1-p_{max}) & -p_{max}^2 & \dots & \vdots \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ \vdots & \cdots & -p_{max}^2 & p_{max}(1-p_{max}) & -p_{max}^2 \\ -p_{max}^2 & \cdots & \cdots & -p_{max}^2 & p_{max}(1-p_{max}) \end{pmatrix}.$$

In a similar manner, we can verify that p_{max} is an eigenvalue with multiplicity $k^* - 1$. The associated eigenvectors are:

$$v_1 = (1, -1, 0, 0, ..., 0, 0), v_2 = (1, 0, -1, 0, ..., 0, 0), ..., v_{k^*-1} = (1, 0, 0, 0, ..., 0, -1)$$

Next, consider the general case for $X_1, X_2, ..., X_m$ with covariance structure Σ_m. Suppose we rearrange $X_1, X_2, ..., X_m$ to be $X_1', X_2', ..., X_{k^*}', X_{k*+1}', ..., X_m'$ so that for $i, j = 1, 2, ..., k^*$,

$$Var(X_j') = p_{max}(1-p_{max}), \qquad Cov(X_i', X_j') = -p_{max}^2$$

Then the covariance structure of $X_1', X_2', ..., X_{k^*}', X_{k*+1}', ..., X_m'$ becomes

$$\Sigma_m' = \begin{pmatrix} \Sigma_{k^*,p_{max}} & B \\ C & D \end{pmatrix}.$$

where $\Sigma_{k^*,p_{max}}$ is defined as above and

$$B_{ij} = -p_{max}p_j, C_{ij} = -p_i p_{max}, D_{ij} = -p_i p_j, D_{ii} = p_i(1-p_i),$$

where $i, j = k^* + 1, ..., m; i \neq j$.

It can be shown that p_{max} is an eigenvalue of Σ_m' with multiplicity $k^* - 1$. The associated eigenvectors are

$$v_1' = (v_1, 0, ..., 0), v_2' = (v_2, 0..., 0), ..., v_{k^*-1}' = (v_{k^*-1}, 0, ..., 0)$$

where $v_1, v_2, ..., v_{k^*-1}$ are defined as above. All other components from k^*+1 to m of $\left(v_j'\right)_{j=1}^{k^*-1}$ are $0s$. In deed, we have $\Sigma_m' v_1' = \begin{pmatrix} \Sigma_{k^*,p_{max}} & B \\ C & D \end{pmatrix} \begin{pmatrix} v_1 \\ \mathbf{0} \end{pmatrix} = \begin{pmatrix} \Sigma_{k^*,p_{max}} v_1 \\ Cv_1 \end{pmatrix}$,

where, $\mathbf{0} = (0, ..., 0) \in \mathbb{R}^{m-k^*}$. Since v_1 is an eigenvector of $\Sigma_{k^*,p_{max}}$ corresponding to eigenvalue p_{max} (we showed this earlier), we get $\Sigma_{k^*,p_{max}} v_1 = p_{max} v_1$. In addition, multiplying C,an $(m-k^*) \times k^*$ matrix, with $v_1 = (1, -1, 0, ..., 0)'$ results:

$$Cv_1 = \begin{pmatrix} -p_{k^*+1}p_{max} & -p_{k^*+1}p_{max} & -p_{k^*+1}p_{max} & \cdots & -p_{k^*+1}p_{max} \\ -p_{k^*+2}p_{max} & -p_{k^*+2}p_{max} & -p_{k^*+2}p_{max} & \cdots & -p_{k^*+2}p_{max} \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ -p_{m-1}p_{max} & \cdots & -p_{m-1}p_{max} & -p_{m-1}p_{max} & -p_{m-1}p_{max} \\ -p_m p_{max} & \cdots & \cdots & -p_m p_{max} & -p_m p_{max} \end{pmatrix} * \begin{pmatrix} 1 \\ -1 \\ 0 \\ \vdots \\ 0 \end{pmatrix}$$

$$= \begin{pmatrix} 0 \\ 0 \\ 0 \\ \vdots \\ 0 \end{pmatrix}.$$

Hence, $\Sigma_m' v_1' = \begin{pmatrix} p_{max} v_1 \\ \mathbf{0} \end{pmatrix} = p_{max} \begin{pmatrix} v_1 \\ \mathbf{0} \end{pmatrix} = p_{max} v_1'$. Here, $\mathbf{0} = (0, ..., 0) \in \mathbb{R}^{m-k^*}$.

Thus, v_1' is an eigenvector of Σ_m' corresponding to eigenvalue p_{max}.

Similarly, we can show that $v_2', ..., v_{k^*-1}'$ are eigenvectors of Σ_m' corresponding to eigenvalue p_{max}.

From Σ_m', we can obtain Σ_m by rearranging the order of the entries. So if we have the same arrangement for $v_1', ..., v_{k^*-1}'$, we will get $v_1'', ..., v_{k^*-1}''$ and the latter are eigenvectors of Σ_m. Their associated eigenvalue is p_{max}. Thus p_{max} is an eigenvalue of Σ_m

with multiplicity $k^* - 1$.

4.3.2 Eigenvalues of Σ_m

Now, let us suppose that $\{p_1, p_2, ..., p_m\}$ can be divided into 2 groups: $G_1 = \{j : p_j \text{ with multiplicity } k_j, k_j > 1\}$ and $G_2 = \{p_j \text{ with multiplicity equal to1}\}$. So Σ_m has eigenvalues 0, $\{p_j, j \in G_1\}$ with multiplicities $\{k_j - 1, j \in G_1\}$ and while the remaining ones are the roots of a polynomial. We still do not have a good idea what such a polynomial looks like. However, if all the ps are distinct, we obtain the following results. Suppose $p_1 \neq p_2 \neq p_3 \neq ... \neq p_m$. Then the eigenvalues are the roots of the following equation

$$x^m + \sum_{j=2}^{m} (-1)^{j-1} j \left(\sum_{i_1 \neq i_2 \neq ... \neq i_j i_1, ... i_j = 1, ..., m} p_{i_1} p_{i_2} ... p_{i_j} \right) x^{m+1-j} = 0.$$

For example, for $m = 3$: if $p_1 \neq p_2 \neq p_3$, then the eigenvalues are the three distinct roots of the polynomial $x^3 - 2(p_1p_2 + p_1p_3 + p_2p_3)x^2 + 3p_1p_2p_3x$.

For $m = 4$: if $p_1 \neq p_2 \neq p_3 \neq p_4$, then the eigenvalues are the four distinct roots of

$$x^4 - 2(p_1p_2 + p_1p_3 + p_1p_4 + p_2p_3 + p_2p_4 + p_3p_4)x^3$$
$$+3(p_1p_2p_3 + p_1p_3p_4 + p_2p_3p_4)x^2 - 4p_1p_2p_3p_4x.$$

For $m = 5$: if $p_1 \neq p_2 \neq p_3 \neq p_4 \neq p_5$, then the eigenvalues are the five distinct roots of

$$x^5 - 2\sum_{i=1}^{4}\sum_{j=i+1}^{5} p_ip_jx^4 + 3\sum_{i=1}^{3}\sum_{j=i+1}^{4}\sum_{k=j+1}^{5} p_ip_jp_kx^3$$
$$-4\sum_{i=1}^{2}\sum_{j=i+1}^{3}\sum_{k=j+1}^{4}\sum_{l=k+1}^{5} p_ip_jp_kp_lx^2 + 5p_1p_2p_3p_4p_5x.$$

4.3.3 Proof that shows multinomial distribution is sub-Gaussian

In the next section, we discuss the procedure to estimate the multiplicity of p_{max}. The key idea comes from (43). To obtain this result, we must show that multinomial distribution is sub-Gaussian. This section is to serve this purpose.

Let $\mathbf{Y} = \mathbf{X} - n\mathbf{p} = (X_1 - np_1, ..., X_m - np_m)$, so

$$\mathbb{E}(Y) = \mathbf{0} \, and \, Cov(\mathbf{Y}) = Cov(\mathbf{X}) = \Sigma_m.$$

It suffices to show $\mathbf{Y}$ is sub-Gaussian, i.e., for all $\mathbf{a} = (a_1, ..., a_m) \in \mathbb{R}^m$, there exist a contant $c > 0$ such that

$$\mathbb{E}(t \sum_{j=1}^{m} a_j Y_j) \leq e^{c^2 t^2}, \text{ for all } t \in \mathbb{R}.$$

Let $A(t) = \mathbb{E}(t \sum_{j=1}^{m} a_j Y_j) = \left(\sum_{j=1}^{m} p_j e^{t a_j} e^{- \sum_{i=1}^{m} t a_i p_i} \right)^n$. So,

$$A^{1/n}(t) = \sum_{j=1}^{m} p_j e^{t a_j} e^{- \sum_{i=1}^{m} t a_i p_i} = \exp\left(\log(\sum_{j=1}^{m} p_j e^{t a_j}) - \sum_{i=1}^{m} t a_i p_i \right).$$

It is enough to show that $A^{1/n}(t) \leq e^{c^2 t^2}$ for all $t \in \mathbb{R}$, or equivalently, $\log A^{1/n}(t) \leq c^2 t^2$ for all $t \in \mathbb{R}$. First,

$$\lim_{t \to 0} \frac{\log(\sum_{j=1}^{m} p_j e^{t a_j}) - \sum_{i=1}^{m} t a_i p_i}{t^2} = \frac{\sum_{j=1}^{m} p_j a_j^2 - (\sum_{i=1}^{m} a_i p_i)^2}{2}.$$

Thus, for every $\epsilon > 0$, there exists a $\delta(a, \epsilon) > 0$ such that $|t| < \delta(a, \epsilon)$ implies

$$\frac{\log(\sum_{j=1}^{m} p_j e^{t a_j}) - \sum_{i=1}^{m} t a_i p_i}{t^2} \leq c(a, p) + \epsilon, \tag{39}$$

where $c(a, p) = \dfrac{\sum_{j=1}^{m} p_j a_j^2 - (\sum_{i=1}^{m} a_i p_i)^2}{2} > 0$ since

$$\left(\sum_{i=1}^{m} a_i p_i \right)^2 \leq \sum_{i=1}^{m} a_i^2 (\sqrt{p_i})^2 \sum_{j=1}^{m} (\sqrt{p_j})^2 = \sum_{i=1}^{m} p_i a_i^2.$$

Let us compute this $\delta(a,\epsilon)$. Using the fact that $\log(1+x) \le x - x^2/2 + x^3/3$ for $x > -1$, we obtain

$$\begin{aligned}\log A^{1/n}(t) &= \log\left[1 + \left(\textstyle\sum_{j=1}^m p_j e^{ta_j} - 1\right)\right] - \textstyle\sum_{i=1}^m ta_i p_i \\ &\le x - \frac{x^2}{2} + \frac{x^3}{3} - \textstyle\sum_{i=1}^m ta_i p_i,\end{aligned}$$

where $x = x(t) = \sum_{j=1}^m p_j e^{ta_j} - 1 > -1$.

Now applying the Taylor expansion for $e^{ta_j}, j = 1, ..., m$, we get

$$\begin{aligned}x(t) &= \textstyle\sum_{j=1}^m p_j e^{ta_j} - 1 \\ &= \textstyle\sum_{j=1}^m p_j \left(1 + ta_j + \frac{t^2}{2}a_j^2 + R_j^{(2)}(t)\right) - 1 \\ &= t\textstyle\sum_{j=1}^m p_j a_j + \frac{t^2}{2}\sum_{j=1}^m p_j a_j^2 + \sum_{j=1}^m p_j R_j^{(2)}(t),\end{aligned}$$

where $|R_j^{(2)}(t)| \le \frac{|a_j|^3}{6} e^{|a_j|}|t|^3$ for $|t| < 1$. Hence, for $|t| < 1$,

$$\begin{aligned}\log A^{1/n}(t) &\le \frac{t^2}{2}\textstyle\sum_{j=1}^m p_j a_j^2 + \sum_{j=1}^m p_j R_j^{(2)}(t) - \frac{x^2}{2} + \frac{x^3}{3} \\ &\le t^2 c(a,p) + |t|^3 b(a,p),\end{aligned}$$

where

$$b(a,p) = b_1 + \frac{1}{2}\left(|b_2|b_3 + b_3 b_1 + 2b_1|b_2|\right) + \frac{1}{3}\left(\frac{3}{2}|b_2| + b_1\right)^3, \tag{40}$$

and where

$$b_1 = b_1(a,p) = \sum_{j=1}^m p_j \frac{|a_j|^3}{6} e^{|a_j|}, b_2 = b_2(a,p) = \sum_{j=1}^m p_j a_j, \text{ and } b_3 = b_3(a,p) = \sum_{j=1}^m p_j a_j^2.$$

Thus, for $|t| < 1$,

$$\frac{\log A^{1/n}(t)}{t^2} \le c(a,p) + |t| b(a,p).$$

So, for each $\epsilon > 0$, there exist $\delta(a,\epsilon) = \min\{\frac{\epsilon}{b(a,p)}, 1\}$ such that for all $|t| < \delta(a,\epsilon)$,

$$\frac{\log A^{1/n}}{t^2} < c(a,p) + \epsilon.$$

Next, we consider when $|t| > \delta(a,\epsilon)$:

$$\begin{aligned}
&\frac{\log(\sum_{j=1}^m p_j e^{ta_j}) - \sum_{i=1}^m ta_i p_i}{t^2} \\
&\leq \frac{\log(\sum_{j=1}^m p_j e^{|t|||a||_\infty}) + \sum_{i=1}^m |t|||a||_\infty p_i}{t^2} \\
&\leq \frac{2|t|||a||_\infty}{t^2} \\
&\leq \frac{2}{\delta(a,\epsilon)}||a||_\infty.
\end{aligned}$$

In summary we have for any fixed $\epsilon > 0$,

$$\frac{\log(\sum_{j=1}^m p_j e^{ta_j}) - \sum_{i=1}^m ta_i p_i}{t^2} \leq \max\left\{c(a,p) + \epsilon, \frac{2}{\delta(a,\epsilon)}||a||_\infty\right\},$$

for all $t \in \mathbb{R}$ and where $\delta(a,\epsilon) = \frac{\epsilon}{b(a,p)}$ and $b(a,p)$ is defined as in (40).

Thus,

$$\mathbb{E}\left(t\sum_{j=1}^m a_j Y_j\right) \leq \exp\left[n\max\left\{c(a,p) + \epsilon, \frac{2b(a,p)}{\epsilon}||a||_\infty\right\}t^2\right],$$

proving the sub-Gaussian requirement.

4.3.4 Estimating k^*

From Section 4.3.1, we have if p_{max} is the maximum probability with multiplicity k^*, then it is an eigenvalue of Σ_m with multiplicity $k^* - 1$. So we can first estimate the multiplicity of the maximum eigenvalue of Σ_m and then add 1 to obtain an estimator for k^*.

Let us consider a preliminary sample: $Y_1, ..., Y_l$ where

$$Y_i = \left(X_1^{(i)}, X_2^{(i)}, ..., X_m^{(i)}\right), i = 1, ..., l. \tag{41}$$

Let $\bar{Y}_l = \frac{1}{l}\sum_{i=1}^{l} Y_i$. Then the empirical estimator for the covariance matrix Σ_m is

$$\hat{\Sigma}_l = \frac{1}{l}\left(Y_i - \bar{Y}_l\right)\left(Y_i - \bar{Y}_l\right)^T \tag{42}$$

As shown in Section 4.3.3, multinomial distribution is sub-Gaussian. Hence, by Corollary 5.50 in [36], we get with probability close to 1 ($\geq 1 - e^{-m\log m}$),

$$||\hat{\Sigma}_l - \Sigma_m||_\infty \leq C\sqrt{\frac{r(\Sigma_m)\log m}{l}}, \tag{43}$$

where C is some constant defined in the corollary, $||.||_\infty$ is spectral norm of the corresponding matrices and

$$r(\Sigma) = \frac{tr(\Sigma_m)}{||\Sigma_m||_\infty} = \frac{\sum_{j=1}^{m} p_j(1-p_j)}{p_{max}} = \frac{1-\sum_{j=1}^{m} p_j^2}{p_max} \leq \frac{1}{p_{max}} \leq m.$$

Thus,

$$||\hat{\Sigma}_l - \Sigma_m||_\infty \leq C\sqrt{\frac{m\log m}{l}}.$$

That is for any $j = m, ..., 1$,

$$|\hat{\lambda}_{(j)} - \lambda_{(j)}| \leq C\sqrt{\frac{m\log m}{l}},$$

where $\hat{\lambda}_{(m)} \geq \hat{\lambda}_{(m-1)} \geq ...\hat{\lambda}_{(1)}$ are eigenvalues of $\hat{\Sigma}_l$ and
$\lambda_{(m)} \geq \lambda_{(m-1)} \geq ... \geq \lambda_{(1)}$ are eigenvalues of Σ_m.

Let $\eta = p_{max} - p_{2nd} > 0$. We can choose l large enough so that

$$2C\sqrt{\frac{m\log m}{l}} < \frac{\eta}{2}.$$

For $\forall j = m, ..., m-(k^*-2)$,

$$|\hat{\lambda}_{(j)} - \hat{\lambda}_{(j-1)}| \leq |\hat{\lambda}_{(j)} - p_{max}| + |p_{max} - \hat{\lambda}_{(j-1)}| \leq 2C\sqrt{\frac{m\log m}{l}} < \frac{\eta}{2}.$$

Hence, by setting $I_j = \left[\hat{\lambda}_{(j)} - C\sqrt{\frac{m\log m}{l}}, \hat{\lambda}_{(j)} + C\sqrt{\frac{m\log m}{l}}\right]$, we have for $j = m, m-1, ...m-(k^*-2)$,

$$I_j \cap I_{j-1} \neq \emptyset.$$

However, for $j = m - (k^* - 1)$,

$$\begin{aligned}
|\hat{\lambda}_{(j)} - \hat{\lambda}_{(j-1)}| &\geq |\lambda_{(j)} - \lambda_{(j-1)}| - |\hat{\lambda}_{(j)} - \lambda_{(j)}| - |\hat{\lambda}_{(j-1)} - \lambda_{(j-1)}| \\
&\geq (p_{max} - p_{2nd}) - 2C\sqrt{\frac{m \log m}{l}} \\
&\geq \eta - \frac{\eta}{2} \\
&\geq 2C\sqrt{\frac{m \log m}{l}}.
\end{aligned}$$

Thus, $I_{m-(k^*-1)} \cap I_{m-(k^*-2)} = \emptyset$.

Let $\hat{k}^*$ be an estimate of k^*, then $\hat{k}^* - 1$ is an estimate of the multiplicity of the largest eigenvalue of Σ_m. The following procedure is used to estimate the multiplicity of largest eigenvalue of Σ_m.

1. Take sample and divide it into l batches $Y_1, ..., Y_l$ of size n as in (41).
2. Compute $\hat{\Sigma}_l$ by (42) and its eigenvalues $\hat{\lambda}_{(m)} \geq \hat{\lambda}_{(m-1)} \geq ...\hat{\lambda}_{(1)}$.
3. Set $I_j = \left[\hat{\lambda}_{(j)} - C\sqrt{\frac{m \log m}{l}}, \hat{\lambda}_{(j)} + C\sqrt{\frac{m \log m}{l}}\right], j = m, m-1, ...1.$
4. Set $j = m$. If $I_j \cap I_{j-1} = \emptyset$, then $\hat{k}^* - 1 = j$. Otherwise, set $j = j - 1$ and repeat step 4.

The estimate of the multiplicity of p_{max} is then $\hat{k}^*$.

4.4 When k^* gets large: Refining Confidence Intervals

Section 4.3 constructs a confidence interval for p_{max} by estimating k^* and the plug the estimator in (38). Here we refine the confidence interval (38) by modifying its upper bound and lower bound. Recall from (38) that

$$\frac{X_{(m)}}{n} - \left(b_{k^*} + \frac{3.676}{a_{k^*}}\right)\frac{\sqrt{X_{(m)}}}{n} \leq p_{max} \leq \frac{X_{(m)}}{n} - \left(b_{k^*} - \frac{1.305}{a_{k^*}}\right)\frac{\sqrt{X_{(m)}}}{n},$$

where $a_{k^*} = (2\log k^*)^{1/2}$ while $b_{k^*} = (2\log k^*)^{1/2} - \frac{1}{2}(2\log k^*)^{-1/2}(\log\log k^* + \log 4\pi)$.

Let us consider the upper bound first:

$$p_{max} \leq \frac{X_{(m)}}{n} - \left(b_{k^*} - \frac{1.305}{a_{k^*}}\right)\frac{\sqrt{X_{(m)}}}{n}.$$

Since this holds true for k^* large, we can suppose $k^* \geq 4$. Hence,

$$p_{max} \leq \frac{X_{(m)}}{n} - \left(b_{k^*} - \frac{1.305}{a_{k^*}}\right)\frac{\sqrt{X_{(m)}}}{n} \leq \frac{X_{(m)}}{n} - \left((2\log 4)^{1/2} - \frac{1.305}{(2\log 4)^{1/2}}\right)\frac{\sqrt{X_{(m)}}}{n}. \tag{44}$$

As for the lower bound:

$$\frac{X_{(m)}}{n} - \left(b_{k^*} + \frac{3.676}{a_{k^*}}\right)\frac{\sqrt{X_{(m)}}}{n} \leq p_{max}$$

$$\iff X_{(m)} - \left(b_{k^*} + \frac{3.676}{a_{k^*}}\right)\sqrt{X_{(m)}} \leq np_{max}.$$

Note that for k^* and m large, we have

$$X_{(m)} - \left(b_m + \frac{3.676}{a_m}\right)\sqrt{X_{(m)}} \leq X_{(m)} - \left(b_{k^*} + \frac{3.676}{a_{k^*}}\right)\sqrt{X_{(m)}} \leq np_{max}.$$

Now for the random sample $(X_1, X_2, ..., X_m)$, let

$$k^1_{ub} = \#\left\{j = 1, ..., m : X_j \geq X_{(m)} - \left(b_m + \frac{3.676}{a_m}\right)\sqrt{X_{(m)}}\right\}.$$

It is clear that $k^* \leq k^1_{ub} \leq m$. Thus, for k^* large, we have

$$X_{(m)} - \left(b_{k^1_{ub}} + \frac{3.676}{a_{k^1_{ub}}}\right)\sqrt{X_{(m)}} \leq X_{(m)} - \left(b_{k^*} + \frac{3.676}{a_{k^*}}\right)\sqrt{X_{(m)}} \leq np_{max}.$$

Next, if we define

$$k^2_{ub} = \#\left\{j = 1, ..., m : X_j \geq X_{(m)} - \left(b_{k^1_{ub}} + \frac{3.676}{a_{k^1_{ub}}}\right)\sqrt{X_{(m)}}\right\},$$

then it can be shown that $k^* \leq k^2_{ub} \leq k^1_{ub} \leq m$. Continuing this process for $i = 3, 4, ..., l$ for some large l, we obtain a decreasing sequence $k^1_{ub}, k^2_{ub}, ..., k^l_{ub}$ such that

$$k^* \leq k^l_{ub} \leq ... \leq k^2_{ub} \leq k^1_{ub} \leq m.$$

Using this technique, we refine the lower bound of the confidence interval for p_{max}. Let us again consider a preliminary sample: $Y_1, ..., Y_l$ where $Y_i = \left(X_1^{(i)}, X_2^{(i)}, ..., X_m^{(i)}\right), i = 1, ..., l$.

Set $i = 1$. Applying the technique above, we are able to obtain k_{ub}^1 :

$$k_{ub}^1 = \#\left\{j = 1, ..., m : X_j^{(1)} \geq X_{(m)}^{(1)} - \left(b_m + \frac{3.676}{a_m}\right)\sqrt{X_{(m)}^{(1)}}\right\}.$$

Then $k^* \leq k_{ub}^1 \leq m$. Next for $i = 2, ..., l$, we define k_{ub}^i as the following:

$$k_{ub}^i = \#\left\{j = 1, ..., m : X_j^{(i)} \geq X_{(m)}^{(i)} - \left(b_{k_{ub}^{i-1}} + \frac{3.676}{a_{k_{ub}^{i-1}}}\right)\sqrt{X_{(m)}^{(i)}}\right\}.$$

As shown above, we will obtain the following sequence $k_{ub}^1, k_{ub}^2, ..., k_{ub}^l$:

$$k^* \leq k_{ub}^l \leq ... \leq k_{ub}^2 \leq k_{ub}^1 \leq m.$$

So by replacing k^* by k_{ub}^l we have a refined lower bound of the confidence interval for p_{max} without knowing k^*:

$$X_{(m)} - \left(b_{k_{ub}^l} + \frac{3.676}{a_{k_{ub}^l}}\right)\sqrt{X_{(m)}} \leq X_{(m)} - \left(b_{k^*} + \frac{3.676}{a_{k^*}}\right)\sqrt{X_{(m)}} \leq np_{max}. \quad (45)$$

Combining (44) and (45), we achieve a confidence interval for p_{max} from sampled data:

$$\boxed{\frac{X_{(m)}}{n} - \left(b_{k_{ub}^l} + \frac{3.676}{a_{k_{ub}^l}}\right)\frac{\sqrt{X_{(m)}}}{n} \leq p_{max} \leq \frac{X_{(m)}}{n} - \left((2\log 4)^{1/2} - \frac{1.305}{(2\log 4)^{1/2}}\right)\frac{\sqrt{X_{(m)}}}{n}.}$$

Remark 4.4.1. *We can also construct a confidence interval by estimating k^* using the technique above. This method is for practical purposes. We will continue to work on proving the theory of this method in the future. Recall that we have constructed a decreasing sequence*

$$k_{ub}^1 \geq k_{ub}^2 \geq ... \geq k_{ub}^{l-1} \geq k_{ub}^l \geq k^* \quad (46)$$

Via simulation, we observe that $(k_{ub}^l)_{l=1}^{\infty}$ *converges to* k^*. *This implies* $(k_{ub}^l)_{l=1}^{\infty}$ *is a Cauchy sequence. Thus, suppose we let* ϵ *to be the acceptable error. That is, there exists an* L *such that for* $l \geq L$,

$$|k_{ub}^l - k^*| < \frac{\epsilon}{2}.$$

Since $(k_{ub}^l)_{l=1}^{\infty}$ *is a Cauchy, there exist an* L' *such that for all* $l, l' \geq L'$,

$$|k_{ub}^l - k_{ub}^{l'}| < \frac{\epsilon}{2}.$$

Using the triangle inequality, we have for $l \geq \max\{L, L'\}$,

$$|k_{ub}^l - k^*| < \epsilon.$$

Hence, given an $\epsilon > 0$, *we can estimate* k^* *and then construct a* 95% *confidence for* p_{max} *using the following procedure:*

1. *From the sample, choose* l *batches of size* n
2. *Construct a decreasing sequence as in (46)*
3. *If*

$$k_{ub}^{l-1} - k_{ub}^l < \frac{\epsilon}{2},$$

   then let k_{ub}^l *be an estimator for* k^*. *Go to step 5.*

4. *Otherwise, obtain a new batch of size* n, *increase* l *to* $l+1$ *and repeat step 3.*
5. *The* 95% *confidence interval for* p_{max} *can be constructed as*

$$\boxed{\frac{X_{(m)}}{n} - \left(b_{k_{ub}^l} + \frac{3.676}{a_{k_{ub}^l}}\right)\frac{\sqrt{X_{(m)}}}{n} \leq p_{max} \leq \frac{X_{(m)}}{n} - \left(b_{k_{ub}^l} - \frac{1.305}{a_{k_{ub}^l}}\right)\frac{\sqrt{X_{(m)}}}{n}}$$

CHAPTER V

APPLICATION TO BIOLOGICAL DIVERSITY MEASUREMENTS

5.1 Introduction

Biological diversity or biodiversity is the degree of variation of life forms within a given species, ecosystem, biome, or an entire planet. Biological diversity is used to describe the variety and variability of life on Earth. Global biodiversity is usually divided into three categories:species diversity, genetic diversity and ecosystem diversity. Species diversity refers to the variety of species within a region; genetic diversity refers to the differences in genetic make-up between distinct species and to generic variations within species; and finally ecosystem diversity refers to the variety of habitats, biotic communities, and ecological processes, as well as the diversity present within ecosystems. Here, our research focuses on species diversity.

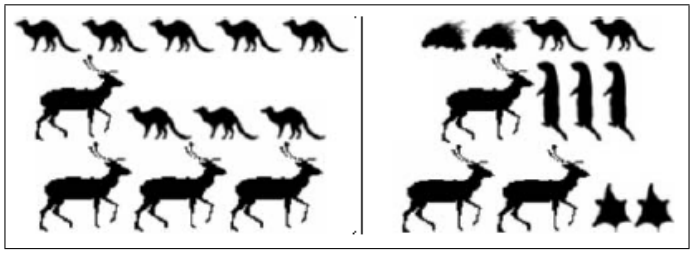
The study of biological diversity has seen a tremendous growth over the past twenty years. Ecologists have always been intrigued by patterns of species abundance and diversity ([32] and [12]). Biological diversity is no longer the sole concern of ecologists and environmentalists. Instead, it has become a matter of public preoccupation and political debate. Many people outside the scientific community are now aware that biodiversity is being eroded at an accelerating rate even if few fully comprehend the magnitude of the loss. No single catalogue of global biodiversity is yet available and estimates of the total number of species on earth vary by an order of magnitude ([25], [26], [27] and [22]). Heightened interest in biodiversity has led to the development of important measurement techniques. Most ecologists recognize two aspects of biodiversity that must be considered when we try to quantify biodiversity: species richness

is the number of species in a community and species evenness measures how the individuals are spread out among the species in a community. There is a perceived need for "indices" of diversity that capture both the richness and evenness characteristics of an ecological community. As there are endless ways of emphasizing different aspects of the species abundance relationship, the number of candidate diversity indices is boundless [28]. Among the commonly used indices, the Berger-Parker index is particularly effective. It provides a simple and easily interpretable measure of dominance ([4] and [24]). The smaller the index, the more diverse (more even) the species is. In this thesis, we develop a statistical method to estimate the proportion of the most abundant species, which relates to the Berger-Parker index, of a community when the number of individuals and the number of species grow without bound. The estimation provides a more accurate diversity comparison among ecological communities.

5.2 Species richness and evenness

Species richness is the measurement counting the number of species found in the observed sample of the community. It is represented by m. When sampling, one takes the individuals one by one, selecting them randomly, and records the species of each individual. At first, new species appears fairly regularly, but as time goes on more and more of the individuals are repeats of species already found. We may never stop finding new species, they just become more and more uncommon.

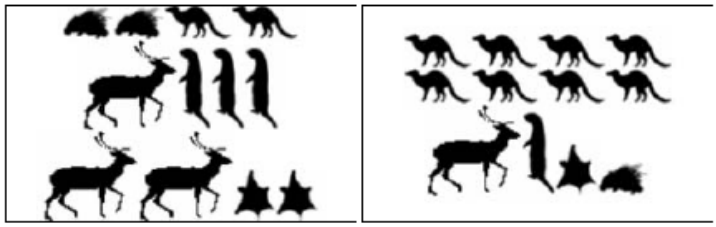
Let us consider an example:



We notice that the community on the left has 2 species and the community on

the right has 5 species. Hence, the community on the right is more diverse.

Species richness measurements provide useful information. Moreover, they have the advantage of being very graphic and easy to understand and can be easily explained to general audiences. But by themselves species richness can be misleading in comparing the diversity of two or more communities. Consider two communities that have the same number of species but one has a very common species with only a few individuals of other species while the other community has more evenly distributed species.



Here, both communities have the same number of species but very intuitively we see that the community on the left is more diverse. We recognize that the community where the individuals are spread out equally among the species is a more diverse community than the one where almost all of the individuals belong to one species.
In most communities some species are going to be common, others less common, and still others rare. However, the extent to which this is true varies from community to community and from situation to situation. There are mathematical formulas for measuring how evenly the individuals are distributed among the species and in some cases they are useful.
One measurement ecologists use to measure the evenness of a population is the Berger-Parker index. In a biological diversity context, the Berger-Parker index d quantifies the proportion of the most abundant species:

$$d = \frac{X_{(m)}}{n},$$

where $X_{(m)}$ is the number of individuals in the most abundant species and n is the total number of individuals sampled.

The smaller the index, the more diverse (more even) the community is. However, when comparing the diversity (evenness) of a community with growing number of individuals and species, the usual way of comparing the ds directly is not accurate any more since d changes its value as n and m grow large. Therefore, a confidence interval would provide a range for p_{max}, which relates to the Berger-Parker index of the given community, and hence helps enhance the diversity comparison.

5.3 Statistical Analysis

We now look at data set, compute the Berger-Parker index and its confidence interval. Then we conclude the diversity comparison using these results.

Data 1: The 'Silhouettes' samples

Table 1: 'Silhouettes' samples

Species	A_1	A_2	B_1	B_2	C_1	C_2	D_1	D_2
Civet	8	3	8	4	2	8	0	0
Sambar	4	2	4	2	3	1	2	3
Porcupine	0	2	0	0	2	1	4	0
Otter	0	3	0	0	3	1	3	0
Colugo	0	2	0	0	2	1	3	0
Mousedeer	0	0	0	0	0	0	0	4
Pig	0	0	0	0	0	0	0	2
Gaur	0	0	0	0	0	0	0	3

Figure 7: Civet;Sambar;Porcupine;Otter;

Figure 8: Colugo;Mousedeer;Pig;Gaur

Notice from the data that

- A_2 has more species than A_1
- B_1 and B_2 each have 2 species, but B_1 has more individuals
- C_1 and C_2 have the same number of species and individuals, but in C_2 most animals are civets
- D_1 and D_2 have the same number of species and individuals, and the proportions are similar

Let us first compare these communities pair-wisely.

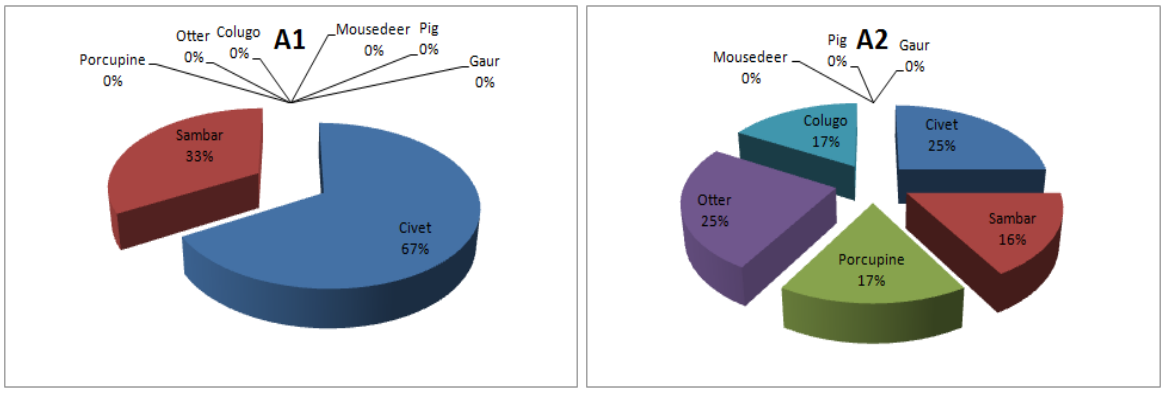


Figure 9: A_2 is more diverse than A_1

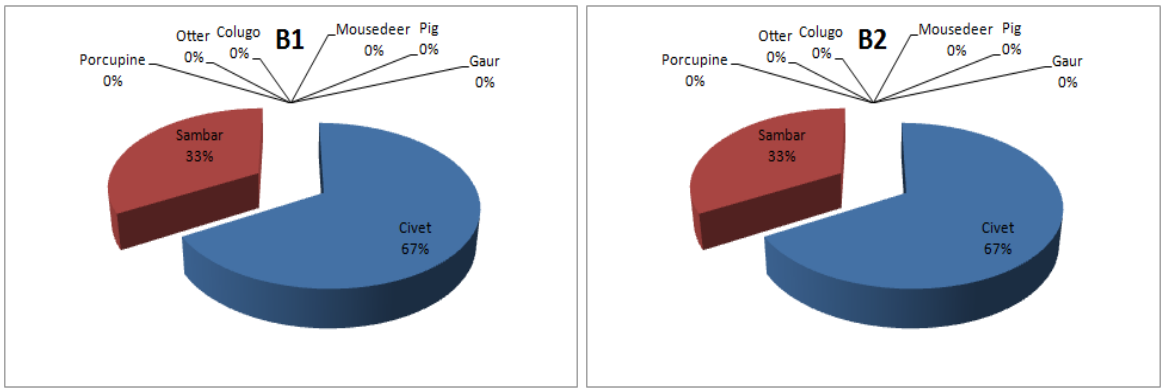


Figure 10: B_1 and B_2 each have 2 species, but B_1 has more individuals. B_1 is as diverse as B2

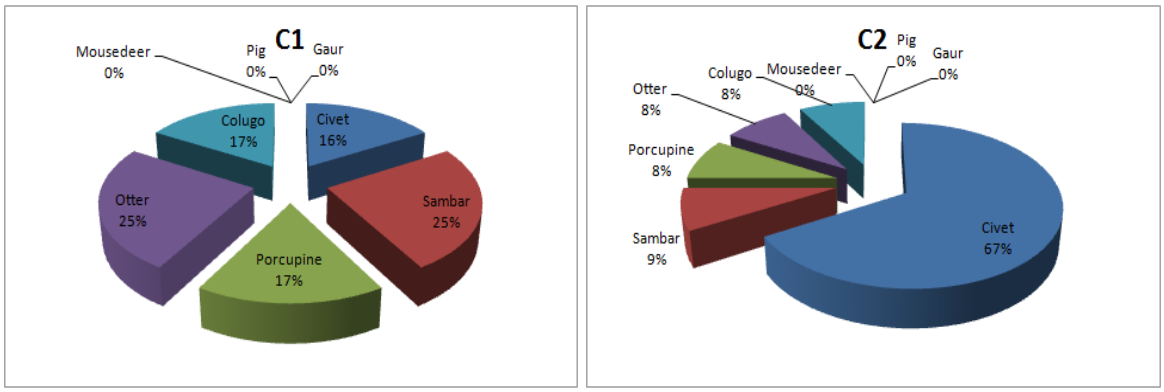


Figure 11: C_1 and C_2 have the same number of species and individuals, but in C_2 most animals are civets. C_1 is more diverse than C_2

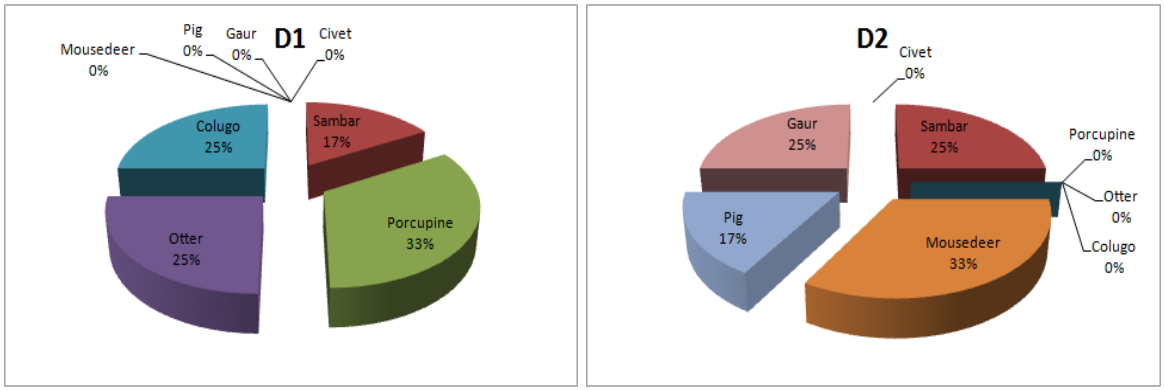


Figure 12: D_1 and D_2 have the same number of species and individuals, and the proportions are similar. They have the same diversity

Now, let us compute the Berger-Parker index of all communities:

Table 2: 'Silhouettes' analysis

	A_1	A_2	B_1	B_2	C_1	C_2	D_1	D_2
No. of individuals (n)	12	12	12	6	12	12	12	12
No. of species (m)	2	5	2	2	5	5	4	4
Berger-Parker index(BPI)	0.67	0.25	0.67	0.67	0.25	0.67	0.33	0.33

We note that from the data analysis, the Berger-Parker index of A_2 and C_1 is the smallest (0.25), so sites A_2 and C_1 are most diverse. However, the sample sizes of all communities are relatively small (≤ 12).

Data 2: Bird abundance data for a range of woodland habitats in County Killarney, Ireland.
The figures represent the number of territories held by breeding males in the respective blocks of habitat. The sites are:

- Oak, Oak2, Oak3 : three oak wood sites
- Yew : a mature yew wood
- Sitka : a Sitka spruce plantation
- Norway : a Norway spruce plantation
- Mixed : a mixed broadleaf wood
- Patchy : a wood with patches of broadleaf and conifer trees
- Swampy : a swampy, seasonally-flooded woodland

Source: Batten, L. A. (1976) Bird communities of some Killarney woodlands, Proceedings of the Royal Irish Academy 76:285-313.
Reference: Magurran 2004 [22] Measuring Biological Diversity, p.237

Using the technique to compute the Berger-Parker index and its 95% confidence interval(CI), we obtain

Table 3: 'Killarney.birds' samples

	Oak	Oak2	Oak3	Yew	Sitka
n	170	182	112	110	75
m	20	22	15	15	8
BPI	0.21	0.23	0.28	0.19	0.40
CI	[.15,.27]	[.17,.29]	[.20,.36]	[.12,.26]	[.29,.51]

	Norway	Mixed	Patchy	Swampy
n	198	91	119	100
m	14	17	21	18
BPI	0.33	0.20	0.18	0.20
CI	[.26,.40]	[.12,.28]	[.11,.25]	[.12,.28]

From the data analysis, we observe that sites Sitka and Norway are least diverse. However, it is not very clear which site is the most diverse.

CHAPTER VI

CONCLUSION

In the thesis, we introduce two estimators for the maximum probability of categorical classes: one associated with the multiomial max and the other derived from the length of the longest increasing subsequence. In both cases, the limiting distribution of the estimators is obtained in Chapters 1 and 2. We then compare the two approaches by looking at their mean square error and bias corrected estimators in Chapter 3. Applications of our work to biological diversity measurement is mentioned in Chapter 5. In particular, the Berger-Parker index, which relates to the maximum proportion of all species, is effective. The results developed in Chapter 1 help estimate the Berger-Parker index as the number of individuals and species simultaneously grows without bound. Constructing the 95% confidence interval for the maximum proportion improves the comparison of Berger-Parker index among communities.

On a short term horizon, we plan to address some of the most interesting open questions resulting from my study of the asymptotics of the multinomial maximum. Here is a sample:

1. Can any of the assumptions (2)-(7) be weakened? In particular, assumptions (3)- (5) come from [37] and conditions (6) and (7) are technical ingredients used in the proof of the theorem.

2. In finding the confidence interval for p_{max}, k^*, the multiplicity of p_{max}, should be estimated. Here we provide two methods of estimating k^*. Are there any better ways estimate k^*?

3. We would like to apply the estimation methodologies developped in this work

to a bacteria population, where there are millions of bacteria. Finding the appropriate data is my challenge at the moment.

4. We would also like to prove that the sequence in (46) actually converges to k^*.

For long term future plans, we believe the framework of estimating the Berger-Parker index for communities with growing number of individuals and species can be applied to other diversity indices. In particular, we will develop the statistical tool for the Shannon index and Simpson index

1. Shannon index
$$H' = \sum_{j=1}^{m} \frac{X_j}{n} \ln\left(\frac{X_j}{n}\right),$$
2. Simpson index
$$D = \sum_{j=1}^{m} \frac{X_j(X_j-1)}{n(n-1)}.$$

APPENDIX A

CODES

The this chapter, we will show all codes that are used through out the thesis. The codes are both in MatLab and R

A.1 Multinomial Maximum

```
r=10000
gamma=0.25
n=10000
m=as.integer(n^gamma)
p=rep(1/m,m)
#p
res=0*(1:r)

am=(2*log(m))^(-0.5)
bm=(2*log(m))^0.5-0.5*(log(log(m))+log(4*pi))/(2*log(m))^0.5

for(i in 1:r){
X=rmultinom(1,n,p)
Y=(X-n/m)/(n/m*(1-1/m))^0.5
res[i]=(max(Y)-bm)/am
#print(res[i])
}
```

```
plot(ecdf(res))


##########3
#Gumble
x=seq(4,14,by=.01)
y=exp(-exp(-(x-6)))
plot(x,y)
```

A.2 Length of the Longest Increasing Subsequence

A.2.1 The length of the longest increasing subsequence LI_n

```
function l=lis(a)
n=length(a);
best=ones(1,n);
prev=ones(1,n);
for i=1:n
    prev(i)=i;
end
for i=2:n
    for j=1:(i-1)
        if ((a(i)>=a(j))&&(best(i)<best(j)+1))
            best(i)=best(j)+1;
            prev(i)=j; %pre[] is for backtracking the sequence
        end
    end
end
```

```
l=max(best);
return
```

A.2.2 Generating LI_n

```
function li=length_lis_general(nRep,n,m,p,numPmax)
m=50;
q=[.01 .02 .03 .03 .05 .05 .05 .05 .01 .02 .03 .03 .01 .02 .03 .03
.01 .02 .03 .03 .01 .02 .03 .03 .01 .02 .03 .03 .01 .02 .03 .03 .01
.02 .03 .03 .01 .02 .03 .03 .01 .02 .03 .03 .05 .05 .05 .05 .05 .05];
p=q-(sum(q)-1)/50;

numPmax=10;

l=zeros(1,nRep);
s=zeros(1,m+1);
s(2)=p(1);
for i=3:(m+1)
    s(i)=s(i-1)+p(i-1);
end

for i=1:nRep
    x=zeros(1,n);
    for k=1:n
        u=rand;
        for j=1:m
            if s(j)<u<s(j+1)
```

```
                x(k)=j;
                break;
            end
        end

    end
    l(i)=lis(x);
end
li=(l-n*max(p))/sqrt(n*max(p));
figure(1);
hist(li,sqrt(nRep));
figure(2);
qqplot(li);
gof=chi2gof(li)
jb=jbtest(li)

return
```

A.2.3 LI_n in the uniform case

```
function l=lis_uniform(nRep,n,m)
for i=1:nRep
    x=unidrnd(m,1,n);
    l(i)=lis(x);
end
l=(l-n/m)/(sqrt(n/m));
figure(1);
```

```
hist(l,sqrt(nRep));
figure(2);
qqplot(l);

return
```

A.2.4 Traceless GUE

```
function mu=GUE_traceless(n,k)
d=zeros(1,n);
for j=1:n
    x= normrnd(0,1,1,k);
    A=zeros(k);
    for r=1:(k-1)
        for s=r:k
            if (r<s)
                y=normrnd(0,sqrt(1/2));
                z=normrnd(0,sqrt(1/2));
                A(r,s)=y+i*z;
                A(s,r)=y-i*z;
            end
            if (r==s)
                A(r,s)=x(r);
            end
        end
    end
   A(k,k)=x(k);
```

```
   B=A-trace(A)/k*eye(k);
   d(j)= max(eig(B));
end
mu=d;

return
```

A.2.5 GUE

```
function GUEt(n,k)
d=zeros(1,n);
a=d;
b=d;
for j=1:n
    x= normrnd(0,1,1,k);
    A=zeros(k);
    for r=1:(k-1)
        for s=r:k
            if (r<s)
                y=normrnd(0,sqrt(1/2));
                z=normrnd(0,sqrt(1/2));
                A(r,s)=y+i*z;
                A(s,r)=y-i*z;
            end
            if (r==s)
                A(r,s)=x(r);
            end
```

```
        end
    end
   A(k,k)=x(k);
   a(j)= max(eig(A));
   B=A-(trace(A)/k)*eye(k);
   b(j)= max(eig(B));
   d(j)= b(j)+normrnd(0,sqrt(1/k));
end

figure(1);
hist(a,sqrt(n));
figure(2);

qqplot(a);

figure(3);
hist(b,sqrt(n));
figure(4);
qqplot(b);

figure(5);
hist(d,sqrt(n));
figure(6);
qqplot(d);

p1=chi2gof(a)
h1=jbtest(a)
```

```
p2=chi2gof(b)
h2=jbtest(b)

p3=chi2gof(d)
h3=jbtest(d)

return
```

A.2.6 Compare $\frac{X_{(m)}}{n}$ and $\frac{LI_n}{n}$

```
function compare_LIS_p_hat(n,nRep)
%We want to use simulation to compare MSE(LI/n) versus MSE(P_hat_max)

p_initial=[.05 .05  .05 .05 .05 .05 .05 .02 .03 .02 .02 .02 .02 .02
.05 .05 .05 .05 .02 .02 .01 .01 .01 .01 .01 .01 .01 .01 .01  .03
.04 .02 .005 .005 .005 .005  .005 .005 .005 .005 .005 .005  .005 .005
.005 .005];
m=length(p_initial);
p_max=max(p_initial);
p_order=randperm(m);
p=zeros(1,m);
for i=1:m
    p(i)=p_initial(p_order(i));
end
```

```
s1=zeros(1,m+1);


s1(2)=p(1);



for i=3:(m+1)
    s1(i)=s1(i-1)+p(i-1);


end
% p_order
% p
sum_p=sum(p)
m


lin=zeros(1,nRep);
Pmax_hat=zeros(1,nRep);
for i=1:nRep
    x=zeros(1,n);
    for k=1:n
        u=rand(1,1);
        for j=1:m
            if ((s1(j)<u)&(u<s1(j+1)))
                x(k)=j;
                break;
            end
         end
```

```
    end
    %This is to get LI/n
    lin(i)=lis(x)/n;

    %Now, this is to get Pmax_hat
    p_hat=zeros(1,m);
    for j=1:m
        for l=1:n
            if (x(l)==j) p_hat(j)=p_hat(j)+1;
            end
        end
    end

    p_hat=p_hat/n;

    Pmax_hat(i)=max(p_hat);
   % x


end

% Pmax_hat

 m1=mean(lin)
 m2=mean(Pmax_hat)
 v1=var(lin)
```

```
 v2=var(Pmax_hat)
 bias1=mean(lin)-p_max
 bias2=mean(Pmax_hat)-p_max

mse1=(mean(lin)-p_max)^2+var(lin)
mse2=(mean(Pmax_hat)-p_max)^2+var(Pmax_hat)
mse11=mean((lin-p_max).^2)
mse21=mean((Pmax_hat-p_max).^2)

k=kstest2(lin,Pmax_hat)
n1=chi2gof(lin)
n2=chi2gof(Pmax_hat)

figure(1);
hist(lin,sqrt(nRep));
figure(2);
qqplot(lin);
figure(3);
hist(Pmax_hat,sqrt(nRep));
figure(4);
qqplot(Pmax_hat);


return
```

A.3 Biological diversity

A.3.1 File 'BiodiversityScript.R'

```
# Downloaded from www.wcsmalaysia.org/stats/ in the file
 Diversity_examples.zip

##############################################################################
#                                                                            #
#                           DIVERSITY INDICES                                #
#                                                                            #
##############################################################################

# This script demonstrates several diversity indices using example data
#    in the R statistical software package.
# If you are not familiar with R, you may want to start by checking out
browseURL("http://www.wcsmalaysia.org/stats/Starting_R.htm")

# You will need the following files, available on the wcsmalaysia.org
#    site in biodiversity.zip:
#   biodiversity.R (has the 'biodiversity' function),
#   Sedilu.trees.R (an example data set),
#   plot.hill.R    (the 'plot.hill' function).

# You will also need the 'vegan' package installed on your computer.
#   If necessary, go to the 'Packages' menu in R, select
#   'Install package(s)...' and follow the on-screen instructions.

# Start R, go to the File menu and select 'Open script...'. Navigate to
```

```
#   this file ("Biodiversity_Script.R") and open it.
# You can then work through the script, using Ctrl-R to run the line
#  where the cursor is positioned.
#  (You don't need to cut-and-paste commands into the R console.)

# ########################################################################

# Load the 'vegan' package, which we will need later:
library(vegan)

# Load the 'biodiversity' function and the example data sets.
# If these files are in the current working directory you can
# run the lines of code below:
source("biodiversity.R")
source("Sedilu.trees.R")
source("plot.hill.R")
# Otherwise, the easiest way is to find these files in My Computer and
# drag-and-drop into the R Console window.

# THE 'SILHOUETTES' SAMPLES
# =========================

# The 'silhouettes' data frame loaded with 'biodiversity' contains the
#   data for the diagrams in:
browseURL("http://www.wcsmalaysia.org/stats/PDF/Diversity_diagrams.pdf")

# Display the data:
```

```
silhouettes

# Notice that:
#   A2 has more species than A1
#   B1 and B2 each have 2 species, but B1 has more individuals
#   C1 and C2 have the same number of species and individuals,
#      but in C2 most animals are civets
#   D1 and D2 have the same number of species and individuals,
#      but in D1 all the species are from different orders, while
#      in D2 they are all ungulates.

# Let's see what the various diversity indices do with this:

biodiversity(silhouettes)

# For details of the various indices:
browseURL("http://www.wcsmalaysia.org/stats/diversityIndexMenagerie.htm")

# Notice that:
#   All the indices show that A2 is more diverse than A1
#   B2 is seen as more diverse than B1 by Simpson, Fisher and Margalef,
#      while Brillouin says B1 is more diverse!
#   C1 is more diverse than C2 for all indices except Fisher and Margalef;
#      all the evenness indices are higher for C1 than C2.
#   D1 and D2 give the same results for all the indices.
```

```
# By default, the function 'biodiversity' calculates a whole series of
#   indices for all columns of the data frame.
# For see more information on the function do this:
cat(comment(biodiversity))

# BORNEAN FOREST TREE DATA
# ========================

# This uses the 'Sedilu.trees' data frame. Look at details of the data set
# with:

cat(comment(Sedilu.trees))

# Before we display the whole data set in the R Console,
check how big it is:

dim(Sedilu.trees)

# 227 rows: that will not be a pretty sight in the R Console window!
# Let's just look at the indices:

biodiversity(Sedilu.trees)

# The two 'Tem.' sites on the hill are more diverse on all measures than
# the rest. The 'Fruit.M' site is also quite diverse. 'Fruit.S' is unusual;
# in fact half the trees in the plot are rubber trees - this must be an
# abandoned rubber garden.
```

```
# Some of the indices give conflicting results, eg. check the values of
# Hill's  N1 and Hill's N2 for 'Sand' and 'Sg.Kara'. How do we decide
# which is "really" more diverse?
# The case is pretty clear cut if ALL Hill's numbers are higher for one
# site than the other. We can plot Hill's numbers with 'plot.hill':

attach(Sedilu.trees)
plot.hill(Fruit.S)

# Let's plot 'Sand' and 'Sg.Kara' on the same graph; they are sites
# 8 and 9:

plot.hill(Sedilu.trees[,8:9])

# The curves cross, so it's not clear which is more diverse. It might
# be more informative to report that "Sandong Paya is richer, but
# Sungai Kara is more even."

# Plotting all the sites together will be messy, but let's try it:

plot.hill(Sedilu.trees)

# Yes, the graph's messy! Only the 'Tem.' sites are clearly more diverse
# than the others.
# Meanwhile, the display in the R Console shows which pairs are clearly
#   different (< or ^) and which are not (.).
```

```
# RANK-ABUNDANCE DIAGRAMS
# ======================

# A good alternative to 'magic number' indices, is to plot
# rank-abundance diagrams.
# This is done using 'as.rad' and 'plot' in the 'vegan' package.

library(vegan)

# Let's try it with the former rubber garden

plot(as.rad(Fruit.S))          # 'as.rad' removes zeros and sorts

# Abundance (as a %age) is plotted on the y axis on a logarithmic scale,
# and rank on the x axis, from most abundant on the left to rarest on the
# right. The rubber trees, with 51% abundance, show up clearly!

# Let's plot some of the high-diversity and low-diversity plots side
# by side:

par(mfrow=c(1,3))
plot(as.rad(Tem.side), main = "Tem.side")
plot(as.rad(Bel.core), main = "Bel.core")
plot(as.rad(Fruit.S), main="Fruit.S")

# When you look at these plots, note that the scales on the y axis are
```

```
#   different.

detach(Sedilu.trees)

# WORKING WITH BASAL AREA
# ======================

# Counting the number of trees for each species might not be the best
# way to measure relative abundance. Species with lots of small trees
# will get higher scores than those with a few big trees. Using Basal
# Area may give a more balanced picture.
# Basal area is the cross-sectional area of the tree trunk, measured
# at a height of 1.5m. The basal areas are then added up for each
# species.

# Basal area data for the Sedilu trees are in the 'Sedilu.BA' data
# frame.

biodiversity(Sedilu.BA)

# Several of the indices can only be used with count data, and they
# are missing from the output from 'biodiversity'.

# The picture looks very different to the results from counts. Now
# 'Bel.core' has the highest diversity on several of the indices.

plot.hill(Sedilu.BA)
```

```
# Now all of the curves cross, none of the sites is clearly more diverse
# than the others. Some are richer, others more even.

# ###########################################################################

# The .zip file contains a data set for birds in County Killarney,
# Ireland. You can use the methods above to investigate the patterns of
# diversity among the different woodland types.

# ###########################################################################
```

A.3.2 File 'biodiversity.R'

```
# Biodiversity and evenness indices, and species richness

"biodiversity" <- function(DATA,
   div.index=c("Chao", "N1", "N2", "BergerParker", "Simpson", "Shannon",
         "Fisher", "Brillouin", "Margalef", "Qstatistic"),
   even.index=c("Hill", "Brillouin", "Simpson", "Shannon"), quiet=FALSE)
   {
   div.index <- tolower(div.index)
   even.index <- tolower(even.index)
   data.name <- deparse(substitute(DATA))
   if(inherits(DATA,"xtabs")) {
      DATA <- as.data.frame.matrix(DATA) # 'as.data.frame.xtabs'
      doesn't do it!!
```

```
} else
   DATA <- as.data.frame(DATA)
# pick out numeric columns without NAs
goodcol <- NULL
for(i in 1:ncol(DATA))
   if(is.numeric(DATA[[i]]) && all(!is.na(DATA[[i]])))
      goodcol <- c(goodcol, i)
if(is.null(goodcol))
   stop("No suitable data found.\n")
DATA <- as.data.frame(DATA[,goodcol])
if(ncol(DATA)==1)
   colnames(DATA) <- data.name

# Check for noninteger values:
integers <- rep(1, ncol(DATA))
integers[!apply(DATA, 2, function(x) all(x == round(x)))] <- NA
# NA indicates non-integers
gotinteger <- sum(integers, na.rm=TRUE) > 0
# TRUE if at least one col is integers

# Number of individuals/column totals and species observed (Sobs):
N <- colSums(DATA)
S <- colSums(DATA > 0)
if(!is.na(sum(integers))) {
   out <- rbind("No. of individuals" = N, "Species observed" = S)
} else
   out <- rbind("Data total" = N, "Species observed" = S)
```

```
# Shannon's H' is used for several calculations:
Shan <- function(data) {
   propi <- data[data > 0]/sum(data)
   return( (-1)*sum(propi * log(propi)) )
}
Shan.H <- apply(DATA, 2, Shan)

if(!is.na(pmatch("c", div.index)) && gotinteger) { # Chao
   F1 <- colSums(DATA==1)
   F2 <- colSums(DATA==2)
   F2[F2==0] <- NA
   chao.cor <- F1^2/(2*F2)
   chao.cor[F1==0] <- 0
   out <- rbind(out, "Chao richness" = S + chao.cor * integers)
}
if(!is.na(pmatch("n1", div.index))) { # Hill's N1
   out <- rbind(out, "Hill\'s N1" = exp(Shan.H) )
}
if(!is.na(pmatch("n2", div.index))) { # Hill's N2
   out <- rbind(out, "Hill\'s N2" =  N^2 / colSums( DATA^2))
}
if(!is.na(pmatch("be", div.index)))  # BergerParker
   out <- rbind(out, "1 / Berger-Parker" = N / apply(DATA, 2, max) )

if(!is.na(pmatch("si", div.index)) && gotinteger)  # Simpson
   out <- rbind(out,
```

```
   "1 / Simpson" = integers*N*(N-1) / colSums( DATA * (DATA-1)))

if(!is.na(pmatch("sh", div.index)))  # Shannon
   out <- rbind(out, "Shannon\'s H\'" = Shan.H )

if(!is.na(pmatch("f", div.index)) && gotinteger) { # Fisher
   nllh <- function(freqs, p, N) {
      i <- as.numeric(names(freqs))
      x <- N/(N + p)
      logExp <- log(p) + log(x) * i - log(i)
      return( sum((log(freqs)-logExp)*freqs - 1) - p*log(1-x) )
   }
   Fish <- function(data) {
      data <- data[data>0]  # remove zeros
      S <- length(data)     # number of species
      freqs <- table(data)
      # convert ni vector into a frequency vector
      N <- sum(data)
      p <- 1/sum((data/N)^2) # a first guess at alpha
      tmp <- nlm(nllh, p=p, freqs=freqs, N = N, hessian = FALSE)
      return(tmp$estimate)
   }
   out <- rbind(out, "Fisher\'s alpha" =
   suppressWarnings(apply(DATA, 2, Fish))*integers)
}
if(!is.na(pmatch("br", div.index)) && gotinteger)  # Brillouin
   out <- rbind(out, "Brillouin" = integers*(lfactorial(N) -
```

```
   colSums(lfactorial(DATA))) / N )

if(!is.na(pmatch("m", div.index)) && gotinteger)  # Margalef
   out <- rbind(out, "Margalef" = integers*(S - 1) / log(N) )

if(!is.na(pmatch("q", div.index))) { # Qstatistic
   Qstat <- function(data) {
      dat <- sort(data[data>0])
      Q1 <- ceiling(length(dat)/4)
      Q3 <- ceiling(3*length(dat)/4)
      Sbq <-  sum(dat>dat[Q1] & dat<dat[Q3]) +
      sum(dat==dat[Q1])/2 + sum(dat==dat[Q3])/2
      Abq <- log(dat[Q3]) - log(dat[Q1])
      return(if (Abq>0) Sbq / Abq else NA)
   }
   out <- rbind(out, "Q-statistic" =  apply(DATA, 2, Qstat))
}

if(!is.na(pmatch("h", even.index)))   # Hill's N2/N1 Evenness
   out <- rbind(out, "Hill's N2/N1 Evenness"
   = N^2 / (colSums(DATA^2)* exp(Shan.H)))

if(!is.na(pmatch("b", even.index)) && gotinteger)
{ # Brillouin evenness
   HB <- (lfactorial(N) - colSums(lfactorial(DATA))) / N
   IntNB <- floor(N / S)
   R <- N - S*IntNB
```

```
      HBmax <- ( lfactorial(N) - lfactorial(IntNB)*(S-R)
      - lfactorial(IntNB+1)*R ) / N
      out <- rbind(out, "Brillouin Evenness" = integers*HB/HBmax )
   }
   if(!is.na(pmatch("si", even.index)) && gotinteger)
   { # Simpson Evenness
      Di <- sweep(DATA*(DATA-1),2,N*(N-1),"/")
      out <- rbind(out, "Simpson Evenness" = integers/( colSums(Di)*S ))
   }
   if(!is.na(pmatch("sh", even.index))) { # Shannon Evenness
      out <- rbind(out, "Shannon Evenness" = Shan.H/log(S) ) # 18 Dec 06
   }



   if(!quiet)  {
      cat("Diversity and Evenness Indices:\n")
      print(round(out, 2), na.print=" ")
      if(is.na(sum(integers)))
         cat("WARNING! Data contains non-integers: some indices could
         not be calculated.\n")
   }
   invisible(out)
}
# ..........................................................

# EXAMPLE DATA
# ============
```

```
# Hypothetical data:
silhouettes <- data.frame(
   A1 = as.integer(c(8,4,0,0,0,0,0,0)),
   A2 = as.integer(c(3,2,2,3,2,0,0,0)),
   # A2 has more species than A1
   B1 = as.integer(c(8,4,0,0,0,0,0,0)),
   B2 = as.integer(c(4,2,0,0,0,0,0,0)),
   # same species count and proportions,
   #but B1 has higher abundance than B2
   C1 = as.integer(c(2,3,2,3,2,0,0,0)),
   C2 = as.integer(c(8,1,1,1,1,0,0,0)),
   # C1 is more even than C2
   D1 = as.integer(c(0,2,4,3,3,0,0,0)),
   D2 = as.integer(c(0,3,0,0,0,4,2,3)),
   # D2 has only ungulates, D1 has a wider range of families
   row.names = c("civet","sambar","porcupine","otter",
   "colugo","mousedeer","pig","gaur")
   )

comment(biodiversity) <-
"
                                ALPHA DIVERSITY

      FUNCTIONS FOR ANALYZING SPECIES RICHNESS, DIVERSITY AND EVENNESS

DESCRIPTION:
```

```
    Computation of a range of diversity and evenness measures from
    species abundance data.

USAGE:

    Biodiversity(DATA, div.index=c(\"Chao\", \"N1\", \"N2\",
    \"BergerParker\", \"Simpson\", \"Shannon\", \"Fisher\",
    \"Brillouin\", \"Margalef\", \"Qstatistic\"),
    even.index=c(\"Hill\", \"Brillouin\", \"Simpson\",
       \"Shannon\"), quiet=FALSE)

ARGUMENTS:

DATA : a vector with species abundance data, one element per species,
       which may contain zeros for missing species (but not NAs); or
       a matrix or data frame with species abundances from different
       samples (sites, quadrats, transects, etc) in the *columns*.
       24 Sept 06: non-numeric columns and columns with NAs in data
       frames are ignored.

div.index : the name or non-ambiguous abbreviation of the diversity
      indices required; the default is to do all of them.

even.index : the name or non-ambiguous abbreviation of the evenness
      indices required; the default is to do all of them.

quiet : if TRUE suppresses output to the Console. Otherwise results
```

are displayed.

VALUE:

A data frame with columns corresponding to those of DATA (one column if DATA is a vector) and the following rows:

No. of individuals* or total data;

Species observed = Hill\'s N0;

Chao\'s estimate of total richness, based on the number of singletons and doubletons (returns NA if the number of doubletons is zero)*;

Hill\'s N1;

Hill\'s N2;

1 / Berger-Parker index of dominance = Hill\'s Ninf;

1 / Simpson\'s index of dominance* = N2 for large samples;

Shannon\'s index (H\') = log(N1);

Fisher\'s alpha parameter for a log series function fitted to the data*;

Brillouin\'s index (appropriate for non-random samples and for complete censuses)*;

Margalef\'s richness index*;

Q-statistic, a measure of the slope of the species abundance curve between the two quartiles;

Hill\'s N2/N1 index of evenness;

Brillouin's index of evenness*;

Simpson\'s index of evenness*;

Shannon\'s index of evenness.

```
     (* These indices are only calculated for integer data.)

Author:

     Mike Meredith; some code based on that by Sam Strindberg, Fernanda
        Marques and Tim O\'Brien of the Wildlife Conservation Society.
        The code for Fisher\'s alpha was inspired by that in the
        \'vegan\' package.

References:

     Magurran, Anne E. (2004) \"Measuring Biological Diversity\",
     Blackwell, Oxford UK
     Hill, M O. (1973) Diversity and evenness: a unifying notation and
     its consequences. Ecology 54:427-431

Examples:

     # Using count data (integers):
     dataset(Killarney.birds)
     biodiversity(Kil.birds[,1:9])
     # Displays the indices with greater precision:
     print(biodiversity(Kil.birds[,1:9], quiet=TRUE))
     # Only individuals, species and the inverse Simpson\'s index:
     biodiversity(Kil.birds[,1:9], div.index=\"si\", even.index=\"\")

     # Using biomass:
```

```
     dataset(woodpeckers)
     biodiversity(Woodpeckers)
     # ...or mixing integer and non-integer data:
     oak.mass <- Kil.birds$Oak * Kil.birds$Bird.mass
     oak.mix <- cbind(oak.count = Kil.birds$Oak, oak.mass)
     biodiversity(oak.mix)

Updated: 19 Nov 06"
```

A.3.3 File 'Killarney.birds.R'

```
# Bird communities of Killarney woodlands (Batten 1976).

Kil.birds <- data.frame(
Oak = as.integer(c(35, 26, 25, 21, 16, 11, 6, 5, 3, 3, 3, 3, 3, 2, 2, 2,
1, 1, 1, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0)),
Oak2 = as.integer(c(41, 35, 19, 27, 8, 9, 4, 8, 3, 2, 8, 3, 3, 0, 1, 1,
0, 2, 1, 1, 0, 0, 0, 0, 1, 1, 3, 1, 0, 0, 0)),
Oak3 = as.integer(c(31, 23, 17, 17, 1, 8, 0, 4, 1, 2, 2, 1, 2, 0, 0, 1,
0, 1, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0)),
Yew = as.integer(c(9, 20, 10, 21, 5, 14, 0, 3, 6, 2, 0, 2,
    9, 6, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 1, 0)),
sitka = as.integer(c(14, 10, 0, 30, 4, 6, 0, 0, 3, 7, 0,
    0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0)),
Norway = as.integer(c(30, 30, 3, 65, 20, 11, 0, 4, 14,
    2, 9, 3, 0, 5, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0,
    0, 0, 1)),
```

```
Mixed = as.integer(c(10, 18, 7, 15, 10, 4, 1,
    2, 7, 3, 2, 1, 4, 3, 0, 1, 1, 0, 0, 0, 0, 0, 0, 0, 2, 0,
    0, 0, 0, 0, 0)),
Patchy = as.integer(c(13, 12, 8, 22, 14,
    6, 3, 3, 8, 5, 2, 1, 3, 3, 1, 0, 2, 0, 0, 0, 5, 1, 0, 2,
    3, 2, 0, 0, 0, 0, 0)),
Swampy = as.integer(c(15, 20, 5, 11,
    10, 5, 0, 2, 8, 4, 3, 3, 2, 6, 0, 0, 1, 1, 0, 0, 2, 1, 1,
    0, 0, 0, 0, 0, 0, 0, 0)),
row.names = c("Chaffinch", "Robin", "BlueTit", "Goldcrest", "Wren",
"CoalTit", "SpottedFlycatcher", "TreeCreeper", "Blackbird", "Siskin",
"Woodpigeon", "LongtailedTit", "GreatTit", "SongThrush", "HoodedCrow",
"Woodcock", "Dunnock", "MistleThrush", "Sparrowhawk", "Redstart",
"WillowWarbler", "Bullfinch", "Crow", "Moorhen", "ChiffChaff", "Mallard",
 "CommonSandpiper", "GreyWagtail", "Jay", "Long-eared owl", "Redpoll"))

comment(Kil.birds) <-
"Data frame \'Kil.birds\' from file \'R\\data\\Killarney.birds.R\'

Bird abundance data for a range of woodland habitats in County Killarney,
   Ireland.

The figures represent the number of territories held by breeding
males in the respective blocks of habitat.

The sites are:
   Oak, Oak2, Oak3 : three oak wood sites,
```

```
   Yew    : a mature yew wood,
   sitka  : a Sitka spruce plantation,
   Norway : a Norway spruce plantation,
   Mixed  : a mixed broadleaf wood,
   Patchy : a wood with patches of broadleaf and conifer trees,
   Swampy : a swampy, seasonally-flooded woodland.

Row names are the English species names.

Source: Batten, L. A. (1976) Bird communities of some Killarney woodlands,
   Proceedings of the Royal Irish Academy 76:285-313.

Reference: Magurran (2004) Measuring Biological Diversity, p237"

#=====================================================================

Bird.mass <- c(21.81, 18.98, 11.88, 5.33, 9.91, 9.06, 14.47, 8.79, 101.8,
12.92, 507.4, 7.78, 18.61, 74.95, 510, 195.6, 21.17, 125.9, 208.6, 14.19,
8.88, 22.51, 508.7, 356.4, 7.7, 1210, 56.28, 18.09, 166.8, 288.8, 10.99)

names(Bird.mass) <- c("Chaffinch", "Robin", "BlueTit", "Goldcrest",
"Wren", "CoalTit", "SpottedFlycatcher", "TreeCreeper", "Blackbird",
"Siskin", "Woodpigeon", "LongtailedTit", "GreatTit", "SongThrush",
"HoodedCrow",  "Woodcock", "Dunnock", "MistleThrush", "Sparrowhawk",
"Redstart",  "WillowWarbler", "Bullfinch", "Crow", "Moorhen",
"ChiffChaff", "Mallard", "CommonSandpiper", "GreyWagtail", "Jay",
"Long-eared owl", "Redpoll")
```

```
comment(Bird.mass) <-
"\'Bird.mass\' contains the approximate weights of individual adult birds,
for the same set of species as in \'Kil.birds\'.

Source: British Trust for Ornithology web site and other sites."

cat("Loaded data frame \'Kil.birds\' and the vector \'Bird.mass\'.\n")
```

A.3.4 File 'Seldilu.trees.R'

```
# Tree data for 11 sites in Sedilu, Sarawak
(Melvin Gumal, unpublished data)

Sedilu.trees <- data.frame(
   Bel.core = c(0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 2, 0, 1, 0, 0, 0, 0, 0,
   0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 7, 0, 3, 0, 0, 0,
   6, 1, 0, 0, 0, 0, 5, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 5, 0, 1, 0, 0,
   7, 0, 0, 0, 0, 0, 0, 0, 0, 78, 0, 0, 0, 0, 0, 0, 0, 9, 0, 0, 0, 0,
   0, 0, 0, 0, 0, 41, 0, 3, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 6, 0, 0, 0,
   0, 0, 0, 3, 0, 0, 0, 3, 0, 4, 4, 0, 0, 0, 0, 0, 0, 1, 0, 0, 5, 0,
   0, 0, 0, 0, 0, 2, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
   0, 0, 1, 0, 0, 0, 0, 0, 0, 4, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0,
   0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 6, 0, 0,
   0, 2, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 3, 19, 6, 1, 0, 1, 1, 0, 0,
   0, 0, 0, 0, 0, 0, 0, 0, 4, 0, 3),
```

```
   Bel.edge = c(0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 2, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 3, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 29, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 38, 0, 2, 0, 0, 0, 0, 0, 0, 0, 2, 0, 0, 12, 0, 0, 0, 0, 0, 0, 0, 0, 0,
1, 2, 0, 0, 0, 3, 0, 8, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 11,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 5, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
11, 0, 0, 0, 6, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 11, 2, 0,
0, 1, 0, 0, 0, 0, 0, 0, 0, 4, 0, 0, 0, 0, 0, 3),
   Bel.roost = c(0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
   0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0,
   0, 0, 0, 0, 0, 18, 0, 0, 0, 0, 0, 1, 0, 0, 3, 0, 0, 0, 0, 0,
   0, 0, 5, 0, 0, 0, 0, 3, 0, 0, 0, 0, 0, 0, 0, 0, 9, 0, 0, 2,
   0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 2, 0, 5, 0, 1, 0, 0, 0,
   0, 0, 0, 0, 1, 0, 0, 8, 0, 0, 0, 0, 0, 0, 3, 0, 0, 0, 2, 0,
   1, 8, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 4, 0, 0, 0, 0, 0, 0, 0,
   0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 2, 0, 1,
   0, 0, 0, 1, 16, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 55, 3, 1,
   0, 0, 0, 0, 0, 5, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
   0, 0, 4, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 3, 9, 0, 0, 1, 5, 0,
   0, 0, 0, 8, 0, 0, 1, 0, 0, 0, 0, 0, 0),


    Fruit.M = c(0, 0, 0, 0, 1, 0, 0, 0, 4, 0, 2, 0, 0, 0,
    2, 0, 0, 1, 1, 7, 3, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
```

```
    0, 0, 0, 0, 0, 0, 0, 0, 0, 2, 0, 0, 0, 0, 1, 0, 0, 0, 0,
    0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 2, 0,
    0, 0, 0, 0, 11, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 1,
    0, 0, 0, 0, 0, 2, 0, 0, 1, 0, 0, 0, 0, 2, 0, 0, 0, 0, 0,
    0, 0, 0, 21, 2, 1, 0, 0, 4, 3, 5, 0, 0, 0, 0, 0, 0, 0, 0,
    0, 0, 0, 0, 0, 0, 0, 1, 0, 1, 0, 0, 2, 0, 0, 0, 0, 1, 3,
    2, 0, 0, 1, 0, 0, 0, 0, 1, 0, 2, 0, 0, 0, 0, 0, 0, 4, 0,
    0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
    0, 1, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 2, 0, 3, 0, 0, 4, 0,
    0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0,
    0, 0, 0, 0),
Fruit.S = c(0, 0, 0, 0, 1, 0, 0, 0, 2, 0,
    0, 0, 0, 0, 6, 1, 1, 0, 0, 1, 0, 0, 0, 0, 0, 1, 0, 0, 0,
    0, 0, 0, 0, 0, 0, 0, 0, 0, 2, 0, 0, 0, 0, 0, 0, 0, 0, 0,
    0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
    0, 0, 0, 0, 0, 0, 0, 0, 3, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0,
    1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0,
    0, 0, 0, 0, 0, 0, 0, 53, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
    0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 4, 0, 0, 0, 0, 0, 0, 0,
    2, 4, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
    0, 0, 2, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 6, 0, 0,
    0, 0, 0, 0, 0, 0, 0, 0, 0, 2, 0, 0, 0, 0, 0, 0, 0, 0, 1,
    0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0,
    0, 0, 0, 1, 6, 0, 0, 0),
Meling = c(0, 3, 0, 0, 0, 0,
    2, 0, 0, 0, 4, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 9, 0,
    0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0,
```

```
    0, 0, 0, 0, 0, 0, 0, 0, 3, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
    1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 5, 0, 0,
    0, 0, 0, 0, 0, 0, 0, 20, 0, 9, 0, 0, 0, 0, 1, 0, 0, 0, 0,
    0, 24, 0, 1, 0, 0, 0, 0, 4, 0, 0, 0, 2, 0, 24, 44, 0, 0,
    0, 0, 0, 0, 0, 0, 0, 14, 1, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0,
    23, 0, 0, 0, 0, 3, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
    0, 0, 0, 0, 0, 0, 3, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
    0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 2, 0, 0, 0,
    0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 2, 0, 0, 0, 0, 0, 0,
    0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0),
Opp.Sand = c(0,
    0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
    0, 0, 0, 2, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
    0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
    0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
    0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 15, 5, 17, 0, 0, 0, 0,
    0, 0, 0, 0, 0, 0, 12, 0, 0, 0, 0, 0, 0, 2, 0, 0, 0, 0, 0,
    16, 20, 0, 0, 0, 0, 0, 0, 0, 0, 38, 0, 11, 0, 0, 0, 0, 0,
    0, 0, 0, 0, 0, 0, 0, 0, 4, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
    0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 7, 0, 0, 0, 0, 0, 0, 0,
    0, 13, 2, 2, 0, 0, 0, 0, 3, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0,
    0, 1, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 4, 2,
    0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 2, 0, 0, 0, 0, 0, 0),
Sand = c(0, 0, 0, 0, 0, 0, 0, 0, 29, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
    0, 0, 0, 0, 4, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 5, 0, 0, 0,
    0, 0, 0, 12, 0, 0, 0, 0, 0, 0, 0, 1, 3, 1, 0, 3, 0, 0, 0,
    0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
```

```
    0, 0, 1, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 1, 0, 0,
    0, 0, 1, 0, 0, 0, 0, 9, 0, 0, 0, 0, 2, 0, 0, 0, 0, 0, 0,
    0, 0, 5, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
    0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0,
    0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
    0, 1, 1, 0, 3, 0, 0, 0, 0, 2, 0, 0, 0, 1, 0, 0, 0, 0, 0,
    1, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1,
    0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 2),
Sg.Kara = c(0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
    0, 0, 0, 0, 2, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
    0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0,
    0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 2, 0, 0, 0, 0,
    0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 13, 5, 10, 0, 0, 0,
    0, 0, 0, 0, 0, 0, 0, 26, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1,
    0, 24, 18, 0, 0, 0, 0, 0, 0, 0, 0, 32, 0, 1, 0, 0, 0, 0,
    0, 0, 0, 0, 0, 0, 0, 0, 0, 2, 0, 0, 0, 0, 0, 0, 3, 0, 0,
    0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 4, 0, 0, 0, 0, 0, 0,
    0, 0, 17, 0, 0, 0, 0, 0, 0, 5, 0, 0, 1, 0, 0, 0, 0, 0, 0,
    0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1,
    2, 0, 0, 0, 0, 0, 0, 1, 0, 2, 0, 1, 0, 0, 0, 0, 0, 0, 0),
Tem.side = c(0, 0, 1, 0, 0, 1, 0, 1, 1, 1, 0, 1, 0, 2,
    0, 0, 0, 0, 1, 0, 1, 0, 1, 0, 1, 0, 0, 0, 3, 2, 0, 0, 0,
    0, 0, 1, 0, 4, 0, 0, 0, 0, 0, 2, 0, 1, 0, 4, 0, 0, 0, 1,
    0, 0, 0, 0, 0, 0, 0, 0, 1, 1, 0, 0, 1, 3, 1, 1, 0, 0, 0,
    0, 2, 0, 0, 0, 2, 0, 0, 0, 0, 1, 0, 2, 0, 1, 2, 0, 2, 0,
    0, 0, 0, 3, 1, 0, 0, 0, 1, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0,
    0, 2, 1, 4, 1, 0, 0, 0, 1, 1, 1, 2, 1, 1, 0, 1, 0, 0, 0,
```

```
    1, 3, 0, 1, 1, 0, 0, 3, 1, 1, 0, 1, 1, 0, 0, 0, 2, 0, 0,
    0, 0, 0, 0, 3, 0, 0, 2, 0, 2, 1, 0, 0, 1, 0, 8, 1, 0, 0,
    0, 2, 1, 1, 2, 0, 0, 0, 0, 0, 1, 1, 0, 0, 0, 2, 0, 1, 0,
    4, 0, 2, 0, 11, 8, 0, 0, 0, 1, 0, 2, 0, 0, 2, 1, 0, 0, 1,
    1, 2, 5, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 1, 0, 0, 1, 0,
    1, 0, 1, 0),
Tem.up = c(1, 1, 0, 7, 0, 0, 0, 0, 1, 0,
    0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 4, 0, 1,
    0, 0, 2, 0, 0, 0, 0, 0, 0, 0, 1, 0, 1, 0, 0, 1, 0, 0, 0,
    1, 0, 0, 0, 0, 0, 1, 2, 0, 0, 3, 0, 0, 1, 0, 0, 0, 0, 0,
    1, 1, 0, 3, 0, 0, 1, 0, 0, 1, 2, 0, 0, 1, 0, 1, 0, 2, 1,
    0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0,
    2, 1, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 2, 7, 0, 0, 0, 0, 0,
    0, 0, 0, 0, 0, 3, 3, 0, 0, 0, 0, 1, 2, 2, 0, 0, 0, 0, 1,
    0, 1, 0, 0, 0, 0, 1, 0, 0, 0, 1, 4, 0, 0, 1, 0, 0, 0, 1,
    4, 0, 0, 1, 0, 0, 0, 1, 0, 0, 0, 0, 1, 0, 0, 1, 0, 0, 0,
    1, 0, 0, 0, 1, 0, 0, 1, 0, 2, 0, 1, 1, 0, 0, 2, 0, 0, 0,
    1, 1, 0, 0, 0, 0, 15, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 2,
    0, 0, 0, 0, 0, 0, 0, 0),


row.names = c("Actinodaphne elegans",
"Actinodaphne myriantha", "Adinandra cordifolia", "Adinandra dumosa",
"Aglaia domestica", "Aglaia tomentosa", "Alangium havilandii",
"Alangium sp.", "Alstonia pneumatophora", "Anisoptera grossivenia",
"Anthocephalus cadamba", "Anthrophyllum diversifolium",
"Aphanomyrtus skiophila",
"Aquilaria microcarpa", "Archidendron jiringa", "Arenga brevipes",
```

"Arenga pinnata", "Arthrophyllum diversifolium",

"Artocarpus aniosophyllus",

"Artocarpus integer", "Artocarpus nitidus", "Artocarpus sesicicarpus",

"Baccaurea angulata", "Baccaurea bracteata", "Baccaurea lanceolata",

"Baccaurea motleyana", "Beilschmiedia tawaensis", "Bhesa paniculata",

"Blumeodendron kurzii", "Brownlowia sp", "Buchanaia sessifolia",

"Callicarpa pentandra", "Calophyllum carum", "Calophyllum hosei",

"Campnosperma coriacea", "Canarium littorale", "Cephalamappa paludicola",

"Chisocheton ceramicus", "Cleistanthus coriaceus",

"Coelostegia griffithii",

"Copaifera palustris", "Cratoxylum arborescens", "Cratoxylum sumatranum",

"Crudia wrayi", "Cryptocarya crassinevia", "Cryptocarya kurzii",

"Cyathocalyx biovulatus", "Dacryodes rostrata", "Dacryodes tapos",

"Dactylocladus stenostachys", "Dehaasia brachybotrys",

"Dehaasia incrassata",

"Dialium laurinum", "Dillenia linn", "Dillenia reticulata",

"Dimocarpus longan",

"Diospyros borneensis", "Diospyros evena", "Diospyros ferrugines",

"Diospyros ferruginescens", "Diospyros lanceifolia", "Diospyros maingayi",

"Diospyros pseudomalabarica", "Diospyros siamang", "Diospyros sp",

"Diospyros wallichii", "Diplopora beccariana", "Dipterocarpus costulatus",

"Dipterocarpus stellatus", "Dracaena cantleyi", "Dryobalanops beccarii",

"Dryobalanops rappa", "Drypetes longfolia", "Drypetes microphylla",

"Durio carinatus", "Durio zibethinus", "Dysoxylum cauliflorum",

"Elaeocarpus cupreus", "Elaeocarpus mastersii", "Elaeocarpus stipularis",

"Elateriospermum tapos", "Endiandra macrophylla", "Endospermum diadenum",

"Erythroxylum cuneatum", "Eugenia alcinae", "Eugenia brachypoda",

"Eugenia castanea", "Eugenia cerina", "Eugenia chrimannii",
"Eugenia havilandii",
"Eugenia palembanica", "Eugenia zeylanica", "Euodia nervosa",
"Euphoria comm.", "Eusideroxylon melangangai", "Eusideroxylon zwageri",
"Evodia nervosa", "Ficus condensa", "Ficus fistulosa", "Ficus sp",
"Ficus treubii", "Ficus uncinata", "Ganua motleyana", "Garcinia beccarii",
"Garcinia blumei", "Garcinia parvifolia", "Gironniera nervosa",
"Glochidion lucidum", "Goniothalamus andersonii", "Gonystylus bancanus",
"Grewia antidesmefolia", "Heritiera albiflora", "Hevea braziliensis",
"Horsfieldia crassifolia", "Hydnocarpus beccariana", "Ilex cymosa",
"Ilex hypoglauca", "Ixonanthes multiflora", "Kibessia korthalsiana",
"Knema cinera", "Knema curtisii", "Knema furfuracea", "Koompassia excelsa",
"Koompassia malaccansis", "Kostermanthus heteropetala",
"Laphopetalum multinervium",
"Licania splendens", "Lindera lucida", "Lithocarpus blumeanus",
"Litsea accedens", "Litsea castanea", "Litsea crassifolia",
"Litsea curtisii",
"Litsea elliptibacea", "Litsea oppositifolia", "Litsea petiolata",
"Litsea resinosa", "Litsea sp", "Lophopetalum multinervium",
"Macaranga aetheadenia", "Macaranga brevipetiolata",
"Macaranga caladiifolia",
"Macaranga hosei", "Macaranga pearsonii", "Macaranga trachyphylla",
"Mangifera indica", "Mangifera pajang", "Mangifera torquenda",
"Melanorrhoea sepciosa", "Meliosma sarawakensis",
"Memmecylon paniculatum",
"Mezzettia havilandii", "Mezzettia leptopoda", "Mezzettia macrocarpa",
"Myristica lowiana", "Myrsine borneensis", "Nauclea gigantea",

"Nauclea peduncularis", "Neonauclea calycina", "Neoscortechinia kingii",
"Neoscortechinia sumatrensis", "Nephelium cuspidatum",
"Nephelium lappaceum",
"Nephelium maingayi", "Nephelium ophiodes", "Nephelium uncinatum",
"Ocalium laurinum", "Ochanostachys amentacea", "Octomeles sumatrana",
"Omphalea bracteata", "Palaquium gutta", "Palaquium walsuraefolium",
"Parastemon urophyllus", "Parishia maingayi", "Parkia singularis",
"Pellacalyx lobbii", "Pentace curtisii", "Pimeleodendron griffithianum",
"Pinanga crassipes", "Platea latifolia", "Polyalthia glauca",
"Polyalthia karai", "Polyalthia sclerophylla", "Pometia acuminata",
"Pometia pannata", "Popowia pisocarpa", "Porterandia anisophylla",
"Prunus arborea", "Prunus turfosa", "Pternandra coerulescens",
"Pterygota horsfieldii", "Quassia indica", "Randia scortechinii",
"Sandoricum borneense", "Sandoricum caudatun", "Sandoricum emarginatum",
"Santiria apiculata", "Santiria tomentosa", "Sarcotheca diversifolia",
"Saurauia glabra", "Scleropyrum wallichianum",
"Scorodocarpus borneensis",
"Semecarpus glaucus", "Shorea collaris", "Shorea dasyphylla",
"Shorea johorensis", "Shorea parvistipulata", "Shorea platycarpa",
"Shorea scaberrima", "Shorea scabrida", "Shorea teysmanniana",
"Simelora jiocarpa", "Stemonurus secundiflorus",
"Sterculia rhodifolia",
"Tabernaemontana macrocarpa", "Teijsmanniodendron hollrungii",
"Terminalia foetidissima", "Tetractomia parviflora",
"Trichadenia philippenensis",
"Untsia sp", "Vatica mangachapoi", "Vatica umbonata", "Vernonia arborea",
"Vitex vestita", "Xanthophyllum amoenum", "Xanthophylum subcorileum",

```
"Xylopia coriifolia"))


# ..............................................................

comment(Sedilu.trees) <-
"Counts of trees >10cm dbh in 11 plots 500m x 5m close to a flying
fox roost  site in Sarawak, Malaysia.

Data are the number of trees of each species in the plot. A few trees
which could not be identified to genus level have been excluded from
the data set.

The plots (columns) are:
   Bel.core  = core Belanga area, Peat Swamp Forest (PSF)
   Bel.edge  = edge of the Belanga area, PSF
   Bel.roost = the flying fox roost site at Belanga, PSF
   Fruit.M   = Fruit-enriched Mixed Dipterocarp Forest (MDF):
       Bukit Melingkong
   Fruit.S   = Fruit-enriched MDF: Sandong
   Meling    = Melingkong paya, PSF
   Opp.Sand  = Opposite Sandong Kecil, PSF
   Sand      = Sandong Paya, logged-over PSF
   Sg.Kara   = Sungai Kara, PSF
   Tem.side  = 50-yr old Temuda (abandoned swidden) on side
               of the hill, Sandong Besar
   Tem.up    = 50-yr old Temuda up hill, Sandong Besar
```

```
The row names are the scientific names of the tree species.

Source: Melvin Gumal, WCS Malaysia, unpublished data.

Updated 12 Dec 06
"

=====================================================
Sedilu.BA <- data.frame(
   Bel.core = c(0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
   0.0342501437533759, 0, 0.0223533117572567,
   0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
   0, 0, 0, 0.333564887479149, 0, 0.0354835945623381, 0,
   0, 0, 0.377650806715604, 0.0198943678864869, 0, 0, 0,
   0, 0.0794421898443196, 0, 0, 0, 0, 0, 0.0108941558546402,
0, 0, 0, 0, 0.288730940010162, 0, 0.0140374659807052, 0, 0,
0.776501051851048, 0, 0, 0, 0, 0, 0, 0, 0, 1.56844809092916,
0, 0, 0, 0, 0, 0, 0, 0.118650010075008, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0.7085259756565, 0, 0.175651352943370, 0, 0, 0, 0, 0,
0, 0, 0.211429384150428,
0, 0, 0.130538884323973, 0, 0, 0, 0, 0, 0, 0.0350936649517629,
0, 0, 0, 0.0751370486336838, 0, 0.068906132611636,
0.06411556882457, 0, 0, 0, 0, 0, 0, 0.350936649517629, 0, 0,
0.588706176749766, 0, 0, 0, 0, 0, 0, 0.03858711595263, 0, 0,
0, 0, 0.0086659866513537, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0.118443108648989, 0, 0, 0, 0, 0, 0, 0.154507618753612, 0,
```

```
0, 0, 0.0108941558546402, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0.458366236104659, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0.159902971324427, 0, 0,
0, 0.0490992999438497, 0.0357223269769759, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0.481204970438346, 0.406370316196536,
0.163253182876512,
0.0223533117572567, 0, 0.0114909868912348, 0.020698100349101,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0.0472133138682107, 0,
0.038626904688403),

   Bel.edge = c(0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0.0692642312335929, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0.105559516005700, 0, 0, 0, 0, 0, 0, 0, 0.662506323861478,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
1.07244966853043, 0, 0.0203797904629172, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0.934876135721793, 0, 0, 0, 0, 0,
0.00764739501556557, 0.03858711595263, 0, 0, 0,
0.0405288062583511, 0, 0.119525362262013, 0.00764739501556557,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0.182614381703641, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0.397282568945989, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0.134621208614280, 0, 0, 0, 0.122199165305957,
0.0198943678864869, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0.704714214769449, 0.453527925834665, 0, 0, 0.0121037334221386,
```

```
0, 0, 0, 0, 0, 0, 0, 0.162783675794391, 0, 0, 0, 0, 0,
0.0813759224028861),

   Bel.roost = c(0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0.0103763619585598, 0, 0, 0, 0, 0, 0, 0, 0,
 0.0112982693537437, 0, 0, 0, 0, 0, 0, 1.03518352988377, 0, 0, 0,
  0, 0, 0.00932185681887703, 0, 0, 0.783809590395587, 0, 0, 0, 0,
0, 0, 0, 0.158362506237564, 0, 0, 0, 0, 0.248923627322090, 0, 0,
0, 0, 0, 0, 0, 0, 0.151197520422399, 0, 0, 0.0927940984859665, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0.0321027770054013, 0,
0.169632529917924, 0, 0.0103763619585598, 0, 0, 0, 0, 0, 0, 0,
0.103386226005101, 0, 0, 0.24036440368459, 0, 0, 0, 0, 0, 0,
0.243693470133752, 0, 0, 0, 0.0259146207281112, 0,
0.0109247987834394, 0.120889128278943, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0.0998931777299021, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0.079804227137315, 0,
0.0571974886033276, 0, 0, 0, 0.0237546113162462,
0.22671389741956, 0, 0, 0, 0, 0, 0, 0.0112982693537437,
0, 0, 0, 0, 1.42527981738293, 0.059824336228073,
0.0103763619585598, 0, 0, 0, 0, 0, 0.134490989217588,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0.0557648052811098,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0.387848402658984, 1.17812939066774,
0, 0, 0.0171859230503057, 0.0757282195475443, 0, 0, 0, 0,
0.148292139074002, 0, 0, 0.00784602038454426, 0, 0, 0, 0, 0, 0),

   Fruit.M = c(0, 0, 0, 0, 0.0616247939651819, 0, 0, 0,
1.42926709319530, 0, 0.0349663409972894, 0, 0, 0, 0.035268735389164,
```

```
0, 0, 0.00814873308630504, 0.0673543719164901, 0.514754832442117,
0.389977357058071, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0.110111347378128, 0, 0, 0, 0, 0.0447623277445956, 0, 0,
0, 0, 0, 0, 0, 0, 0.0127323954473516, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0.0490595112080767, 0, 0, 0, 0, 0, 3.35323549600314, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0.103418882021114, 0.0911082471729555,
0, 0, 0, 0, 0, 0.0282181714101930, 0, 0, 0.0114909868912348, 0, 0,
0, 0, 0.0370831017404116, 0, 0, 0, 0, 0, 0, 0, 0, 1.23906897845333,
 0.0438153558331988, 0.0114909868912348,
0, 0, 0.360883833460873, 0.0515264128260011, 0.0905909936079068,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0.00814873308630504,
0, 0.0673543719164901, 0, 0, 0.0648158505741744, 0, 0, 0, 0,
0.130379729380881, 0.766211684780157, 0.171354169479889, 0, 0,
0.0114909868912348, 0, 0, 0, 0, 0.0097482402643786,
0, 0.0903761344347328, 0, 0, 0, 0, 0, 0, 0.356005734455106, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0.0086659866513537,
0, 0, 0, 0.0367966228428462, 0, 0, 0, 0, 0, 0, 0.192832129050140,
0, 0.0538023285122152, 0, 0, 0.0671315549961615, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0.385154962282387, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0),
   Fruit.S = c(0, 0, 0, 0, 0.0811769787240212,
0, 0, 0, 0.102877755214601, 0, 0, 0, 0, 0, 0.114106136449734,
0.350936649517629, 0.0114909868912348, 0, 0, 0.00814873308630504,
0, 0, 0, 0, 0, 0.0267698614280568, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0.0600014135456446, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0.809095984196268, 0, 0, 0, 0, 0, 0, 0.0191065509181820,
```

```
0, 0, 0, 0.0133769729668738, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0.176669944579158, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
3.81297046461839, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0.198227481620956, 0, 0, 0, 0, 0, 0, 0,
0.0195601425059939, 0.043536834682788, 0.122358320249049, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0.120400714449019, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0.00919915571071155, 0, 0,
0.0774050065727433, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0.02435070629306,
0, 0, 0, 0, 0, 0, 0, 0, 0.0147138744888457, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0.00764739501556557, 0, 0, 0, 0, 0, 0, 0,
0.049664299991826, 0.125764236031216, 0, 0, 0),

   Meling = c(0, 0.134493884659806, 0, 0, 0, 0, 0.0701873299035258,
0, 0, 0, 0.0729645836604794, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0.272385727354624, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0.0240721851426492, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0.0987795154299849, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0.0412529612494193,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0.241613119107806,
0, 0, 0, 0, 0, 0, 0, 0, 0, 1.28318672867841, 0, 0.281314319662080,
0, 0, 0, 0, 0.0086659866513537, 0, 0, 0, 0, 0, 1.25338496558445,
0, 0.0296107771622471, 0, 0, 0, 0, 0.0915618387607674, 0, 0,
0, 0.101859163578813, 0, 0.632585194560202, 1.89317192131956,
0, 0, 0, 0, 0, 0, 0, 0, 0, 2.37962900487994, 0.0086659866513537,
0, 0, 0, 0, 0, 0.00814873308630504, 0, 0, 0, 0, 1.29494827897290,
0, 0, 0, 0, 0.132011067547573, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0.0749619781962827, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0.0175786634644998, 0,
```

```
0, 0, 0, 0, 0, 0, 0, 0.0262446501158535, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0.05017359580972, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0.0389929610575144, 0, 0, 0, 0, 0, 0),
   Opp.Sand = c(0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0.10708248620826, 0, 0, 0, 0.0224090188202969, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0.505176222285382,
0.227664050488128, 0.350624528148591, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0.491062016215625, 0, 0, 0, 0, 0, 0, 0.100833915574009,
0, 0, 0, 0, 0, 0.704208575186231, 0.538826234510615, 0, 0, 0,
0, 0, 0, 0, 0, 1.32815000803246, 0, 0.331644220236340, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0.160496623782160, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0.0120640409432752, 0,
0, 0, 0, 0.270497044971432, 0, 0, 0, 0, 0, 0, 0, 0,
0.58327315538702, 0.0467285436042302, 0.0506413539700025, 0,
0, 0, 0, 0.0811670808781103, 0, 0, 0.0538582223276656, 0, 0, 0,
0, 0, 0, 0, 0, 0.0715588443151974, 0, 0, 0, 0.0425952600656523,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0.137758072105713,
0.0575372212859784, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0.0263759667267224, 0, 0, 0, 0, 0, 0),


   Sand = c(0,
0, 0, 0, 0, 0, 0, 0, 2.01759925582880, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0.058362117631798, 0, 0, 0, 0, 0, 0,
0.0401150034063122, 0, 0, 0, 0.528824129411441, 0, 0, 0, 0, 0,
0, 0.227687061587265, 0, 0, 0, 0, 0, 0, 0, 0.0121037334221386,
```

```
0.0403378203266409, 0.0168385929791225, 0, 0.112721488444835,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0.0168385929791225, 0.0114909868912348, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0.134485926912652, 0, 0, 0.0223533117572567, 0, 0,
0, 0, 0.0147138744888457, 0, 0, 0, 0, 0.429264754760306, 0, 0,
0, 0, 0.0294138229201709, 0, 0, 0, 0, 0, 0, 0, 0,
0.127562686888154, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0.0154061984912955, 0, 0, 0, 0,
0, 0, 0, 0, 0.0183346494441863, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0.0447623277445956, 0.179049310978382, 0, 0.0309556364313736,
0, 0, 0, 0, 0.0303110589118515, 0, 0, 0, 0.158207971180499, 0,
0, 0, 0, 0, 0.0183346494441863, 0, 0, 0, 0, 0, 0, 0,
0.0114909868912348, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0.00814873308630504, 0, 0, 0, 0, 0.0108941558546402, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0.0877262046322527),

   Sg.Kara = c(0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0.0303110589118515, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0.0114909868912348, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0.0631288081774003, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0.546920046440989, 0.370647989219560,
0.335888549648291, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 1.25336109234298, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0.0168385929791225,
0, 0.852354297728646, 0.360716720770626, 0, 0, 0, 0, 0, 0, 0,
```

```
0, 1.39091870965731, 0, 0.0223533117572567, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0.0430514121063577, 0, 0, 0, 0, 0, 0,
0.0623012024733224,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0.136451490459836,
0, 0, 0, 0, 0, 0, 0, 0, 0.822974295233882, 0, 0, 0, 0, 0, 0,
0.146740857530728, 0, 0, 0.0616247939651819, 0, 0, 0, 0, 0, 0,
0, 0, 0.101612473417021, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0.0877341623794073, 0.0206423961190188, 0, 0, 0, 0,
0, 0, 0.0764262036727281, 0, 0.0242711288215140, 0,
0.0168385929791225, 0, 0, 0, 0, 0, 0, 0),

   Tem.side = c(0, 0, 0.0147138744888457,
0, 0, 0.0140374659807052, 0, 0.049664299991826, 0.153751632773925,
0.0121037334221386, 0, 0.0459639475649394, 0, 0.0804846547215715,
0, 0, 0, 0, 0.0484149336885546, 0, 0.120392756701864, 0,
0.0191065509181820, 0, 0.0147138744888457, 0, 0, 0,
0.358504467061649, 0.150759519843798, 0, 0, 0, 0, 0,
0.0522107790812963, 0, 0.0734181752482913, 0, 0, 0, 0, 0,
0.0397887357729738, 0, 0.0191065509181820, 0, 0.104509093381293,
0, 0, 0, 0.00764739501556557, 0, 0, 0, 0, 0, 0, 0, 0,
0.0232047907027983, 0.0108941558546402, 0, 0, 0.0114909868912348,
0.205325792082854, 0.0412529612494193, 0.257831007808870, 0, 0, 0, 0,
 0.0213983820987053, 0, 0, 0, 0.0305975378094169, 0, 0, 0, 0,
0.0183346494441863, 0, 0.0326267633338386, 0, 0.0267698614280568,
0.0860789509712516, 0, 0.0729884569019432, 0, 0, 0, 0,
0.120146066540072, 0.0086659866513537, 0, 0, 0, 0.0223533117572567,
0, 0, 0.0133769729668738, 0, 0, 0, 0, 0, 0, 0, 0, 0.0317195801582147,
```

```
 0.0588554979553829, 0.360748551759245,
0.00814873308630504, 0, 0, 0, 0.0367966228428462, 0.0367966228428462,
0.0548209201480033, 0.0401866231307036, 0.0240721851426492,
0.0574947231919472, 0, 1.00287508515781, 0, 0, 0, 0.0114909868912348,
 0.0757975416475152,
0, 0.0688265551400901, 0.0688265551400901, 0, 0, 0.136984659519194,
0.249308260606299, 0.0928191628111934, 0, 0.0133769729668738,
0.00814873308630504, 0, 0, 0, 0.0378470454672527, 0, 0, 0, 0,
0, 0, 0.027708875592299, 0, 0, 0.173805155603504, 0,
0.0347037353411878, 0.149358956344589, 0, 0, 0.136562898920001, 0,
0.146947758956747, 0.0175786634644998, 0, 0, 0, 0.406044048563198,
0.214031567469981, 0.0175786634644998, 0.0454705672413545, 0, 0, 0,
0, 0, 0.0232047907027983, 0.0277009178451444, 0, 0, 0,
0.0210728306626109, 0, 0.188725931518369,
0, 0.0554814131618347, 0, 0.147218322360003, 0, 0.188630438552514,
0.486083069444112, 0, 0, 0, 0.0535078918674952, 0, 0.101095219851972,
0, 0, 0.0481921167682259, 0.0286478897565412, 0, 0, 0.0147138744888457,
0.825091055977004, 0.0464254968999059, 3.79949004093851, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0.257831007808870, 0, 0.410054753129114,
0, 0, 0.0086659866513537, 0, 0.0877341623794073,
0, 0.0191065509181820, 0),


   Tem.up = c(0.0367966228428462, 0.0168385929791225, 0,
0.377276792599338, 0, 0, 0, 0, 0.0389929610575144, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0.0424068345868355, 0, 0, 0, 0,
0.160666915051268, 0, 0.0121037334221386, 0, 0, 0.0472769758454475,
0, 0, 0, 0, 0, 0, 0, 0.0232047907027983, 0, 0.0844237395630959, 0, 0,
```

```
 0.0191065509181820,
0, 0, 0, 0.145029941892490, 0, 0, 0, 0, 0, 0.0103132403123548,
0.0157961281018706, 0, 0, 0.08728852853875, 0, 0, 0.00814873308630504,
0, 0, 0, 0, 0, 1.61144379880544, 0.00814873308630504, 0,
0.461151447608767, 0, 0, 0.00764739501556557, 0, 0, 0.0086659866513537,
 0.0589828219098564, 0, 0, 0.554623145686637, 0, 0.0249554950768092, 0,
  0.382871088849018, 0.0086659866513537, 0, 0, 0.0305895800622623, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0.00919915571071155, 0, 0, 0, 0,
0.0569933851212077, 0.0215177483060243, 0.0103132403123548, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0.497072718264608, 0.112037122189540, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0.0309556364313736, 0.156632337243889, 0, 0, 0, 0,
0.0764262036727281, 0.307049673960039, 0.0346082423753326, 0,
0, 0, 0, 0.0522107790812963, 0, 0.0114909868912348, 0, 0, 0,
0, 0.020698100349101, 0, 0, 0, 0.0175786634644998, 0.264205163279701,
0, 0, 0.0277009178451444, 0, 0, 0, 0.0198943678864869, 0.284386010063753,
0, 0, 0.00814873308630504, 0, 0, 0, 0.0336214817281629, 0, 0,
0, 0, 0.049664299991826, 0, 0, 0.0140374659807052, 0, 0, 0,
0.0127323954473516, 0, 0, 0, 0.00814873308630504, 0, 0,
0.0435766234185609, 0, 0.265979740895176,
0, 0.0097482402643786, 0.0114909868912348, 0, 0, 0.0302076081988417,
0, 0, 0, 0.0086659866513537, 0.0127323954473516, 0, 0, 0, 0,
1.46852265990892, 0, 0, 0, 0, 0, 0, 0, 0.0161144379880544, 0,
0, 0, 0.109570220571615, 0, 0, 0, 0, 0, 0, 0, 0),

row.names = c("Actinodaphne elegans",
"Actinodaphne myriantha", "Adinandra cordifolia", "Adinandra dumosa",
"Aglaia domestica", "Aglaia tomentosa", "Alangium havilandii",
```

"Alangium sp.", "Alstonia pneumatophora", "Anisoptera grossivenia",
"Anthocephalus cadamba", "Anthrophyllum diversifolium",
"Aphanomyrtus skiophila",
"Aquilaria microcarpa", "Archidendron jiringa", "Arenga brevipes",
"Arenga pinnata", "Arthrophyllum diversifolium",
"Artocarpus aniosophyllus",
"Artocarpus integer", "Artocarpus nitidus", "Artocarpus sesicicarpus",
"Baccaurea angulata", "Baccaurea bracteata", "Baccaurea lanceolata",
"Baccaurea motleyana", "Beilschmiedia tawaensis", "Bhesa paniculata",
"Blumeodendron kurzii", "Brownlowia sp", "Buchanaia sessifolia",
"Callicarpa pentandra", "Calophyllum carum", "Calophyllum hosei",
"Campnosperma coriacea", "Canarium littorale", "Cephalamappa paludicola",
"Chisocheton ceramicus", "Cleistanthus coriaceus",
"Coelostegia griffithii",
"Copaifera palustris", "Cratoxylum arborescens", "Cratoxylum sumatranum",
"Crudia wrayi", "Cryptocarya crassinevia", "Cryptocarya kurzii",
"Cyathocalyx biovulatus", "Dacryodes rostrata", "Dacryodes tapos",
"Dactylocladus stenostachys", "Dehaasia brachybotrys",
"Dehaasia incrassata",
"Dialium laurinum", "Dillenia linn", "Dillenia reticulata",
"Dimocarpus longan",
"Diospyros borneensis", "Diospyros evena", "Diospyros ferrugines",
"Diospyros ferruginescens", "Diospyros lanceifolia", "Diospyros maingayi",
"Diospyros pseudomalabarica", "Diospyros siamang", "Diospyros sp",
"Diospyros wallichii", "Diplopora beccariana", "Dipterocarpus costulatus",
"Dipterocarpus stellatus", "Dracaena cantleyi", "Dryobalanops beccarii",
"Dryobalanops rappa", "Drypetes longfolia", "Drypetes microphylla",

"Durio carinatus", "Durio zibethinus", "Dysoxylum cauliflorum",
"Elaeocarpus cupreus", "Elaeocarpus mastersii", "Elaeocarpus stipularis",
"Elateriospermum tapos", "Endiandra macrophylla", "Endospermum diadenum",
"Erythroxylum cuneatum", "Eugenia alcinae", "Eugenia brachypoda",
"Eugenia castanea", "Eugenia cerina", "Eugenia chrimannii",
"Eugenia havilandii",
"Eugenia palembanica", "Eugenia zeylanica", "Euodia nervosa",
"Euphoria comm.", "Eusideroxylon melangangai", "Eusideroxylon zwageri",
"Evodia nervosa", "Ficus condensa", "Ficus fistulosa", "Ficus sp",
"Ficus treubii", "Ficus uncinata", "Ganua motleyana", "Garcinia beccarii",
"Garcinia blumei", "Garcinia parvifolia", "Gironniera nervosa",
"Glochidion lucidum", "Goniothalamus andersonii", "Gonystylus bancanus",
"Grewia antidesmefolia", "Heritiera albiflora", "Hevea braziliensis",
"Horsfieldia crassifolia", "Hydnocarpus beccariana", "Ilex cymosa",
"Ilex hypoglauca", "Ixonanthes multiflora", "Kibessia korthalsiana",
"Knema cinera", "Knema curtisii", "Knema furfuracea", "Koompassia excelsa",
"Koompassia malaccansis", "Kostermanthus heteropetala",
"Laphopetalum multinervium",
"Licania splendens", "Lindera lucida", "Lithocarpus blumeanus",
"Litsea accedens", "Litsea castanea", "Litsea crassifolia",
"Litsea curtisii",
"Litsea elliptibacea", "Litsea oppositifolia", "Litsea petiolata",
"Litsea resinosa", "Litsea sp", "Lophopetalum multinervium",
"Macaranga aetheadenia", "Macaranga brevipetiolata",
"Macaranga caladiifolia",
"Macaranga hosei", "Macaranga pearsonii", "Macaranga trachyphylla",
"Mangifera indica", "Mangifera pajang", "Mangifera torquenda",

"Melanorrhoea sepciosa", "Meliosma sarawakensis",

"Memmecylon paniculatum",

"Mezzettia havilandii", "Mezzettia leptopoda", "Mezzettia macrocarpa",

"Myristica lowiana", "Myrsine borneensis", "Nauclea gigantea",

"Nauclea peduncularis", "Neonauclea calycina", "Neoscortechinia kingii",

"Neoscortechinia sumatrensis", "Nephelium cuspidatum",

"Nephelium lappaceum",

"Nephelium maingayi", "Nephelium ophiodes", "Nephelium uncinatum",

"Ocalium laurinum", "Ochanostachys amentacea", "Octomeles sumatrana",

"Omphalea bracteata", "Palaquium gutta", "Palaquium walsuraefolium",

"Parastemon urophyllus", "Parishia maingayi", "Parkia singularis",

"Pellacalyx lobbii", "Pentace curtisii", "Pimeleodendron griffithianum",

"Pinanga crassipes", "Platea latifolia", "Polyalthia glauca",

"Polyalthia karai", "Polyalthia sclerophylla", "Pometia acuminata",

"Pometia pannata", "Popowia pisocarpa", "Porterandia anisophylla",

"Prunus arborea", "Prunus turfosa", "Pternandra coerulescens",

"Pterygota horsfieldii", "Quassia indica", "Randia scortechinii",

"Sandoricum borneense", "Sandoricum caudatun", "Sandoricum emarginatum",

"Santiria apiculata", "Santiria tomentosa", "Sarcotheca diversifolia",

"Saurauia glabra", "Scleropyrum wallichianum",

"Scorodocarpus borneensis",

"Semecarpus glaucus", "Shorea collaris", "Shorea dasyphylla",

"Shorea johorensis", "Shorea parvistipulata", "Shorea platycarpa",

"Shorea scaberrima", "Shorea scabrida", "Shorea teysmanniana",

"Simelora jiocarpa", "Stemonurus secundiflorus",

"Sterculia rhodifolia",

"Tabernaemontana macrocarpa", "Teijsmanniodendron hollrungii",

```
"Terminalia foetidissima", "Tetractomia parviflora",
"Trichadenia philippenensis",
"Untsia sp", "Vatica mangachapoi", "Vatica umbonata", "Vernonia arborea",
"Vitex vestita", "Xanthophyllum amoenum", "Xanthophylum subcorileum",
"Xylopia coriifolia"))


comment(Sedilu.BA) <-
"Basal area (sq. m) of trees >10cm dbh in 11 plots 500m x 5m.
A few trees which could not be identified to genus level have
been excluded from the data set.

The plots (columns) are:
   Bel.core  = core Belanga area, Peat Swamp Forest (PSF)
   Bel.edge  = edge of the Belanga area, PSF
   Bel.roost = the flying fox roost site at Belanga, PSF
   Fruit.M   = Fruit-enriched Mixed Dipterocarp Forest (MDF):
                                              Bukit Melingkong
   Fruit.S   = Fruit-enriched MDF: Sandong
   Meling    = Melingkong paya, PSF
   Opp.Sand  = Opposite Sandong Kecil, PSF
   Sand      = Sandong Paya, logged-over PSF
   Sg.Kara   = Sungai Kara, PSF
   Tem.side  = 50-yr old Temuda (abandoned swidden) on side of
               the hill, Sandong Besar
   Tem.up    = 50-yr old Temuda up hill, Sandong Besar
```

The row names are the scientific names of the tree species.

Source: Melvin Gumal, WCS Malaysia, unpublished data.

Updated 12 Dec 06"

REFERENCES

[1] Alam, K. and Feng, Z., "Estimating probability of occurrence of the most likely multinomial event," *JSPI*, vol. 59, no. 2, pp. 257–277, 1997.

[2] Barton, D. and David, F., "Combinatorial extreme value distributions," *Mathematika*, vol. 6, pp. 63–76, 1959.

[3] Baryshnikov, Y., "Gues and queues," *Probab. Theory Related Fields*, vol. 119, no. 2, 2001.

[4] Berger, W. and Parker, F., "Diversity of planktonic foraminifera in deep sea sediments," *Science*, 1970.

[5] Bod, R., *Beyond Grammar. An Experience-Based Theory of Language*. Center for the Study of Language and Information-Stanford, California., 1998.

[6] Breton, J. and Houdré, C., "Asymptotics for random young diagrams when the word length and alphabet size simultaneously grow to infinity," *Bernoulli*, vol. 16, no. 2, 2010.

[7] Diaconis, P. and Graham, R., "The analysis of sequential experiments with feedback to subjects," *Ann. Statist.*, vol. 9, pp. 3–23, 1981.

[8] Ethier, S., "Testing for favorable numbers on a roulette wheel," *J.Amer.Stat.Assoc*, vol. 77, no. 379, pp. 660–665, 1982.

[9] Freeman, P., "Exact distribution of the largest multinomial frequency," *Appl.Statist.*, vol. 28, no. 3, pp. 333–336, 1979.

[10] Good, I., "Saddle-point methods for the multinomial distribution.," *Ann.Math.Statist.*, vol. 28, pp. 861–881, 1957.

[11] Greenwood, R. and Glasglow, M., "Distribution of maximum and minimum frequencies in a sample drawn from a multinomial distribution," *Ann.Math.Statist.*, vol. 21, pp. 416–424, 1950.

[12] Hawkins, B., "Ecology's oldest pattern," *Trends Ecol. Evol*, 2001.

[13] Houdré, C. and Litherland, T., "On the longest increasing subsequence for nite and countable alphabets," *High Dimensional Probability V: The Luminy Volume*, vol. 5, pp. 185–212, 2009.

[14] Its, A., Tracy, C., and Widom, H., "Random words, toeplitz determinants and integrable systems. i," *Random Matrices and their Applications*, pp. 245–258, 1999.

[15] Johnson, N., "An approximation to the multinomial distribution, some properties, and applications," *Biometrika*, vol. 47, pp. 93–102, 1960.

[16] Johnson, N. and Kotz, S., *Discrete Distributions*. NewYork: Houghton Mifflin, 1969.

[17] Johnson, N. and Young, D., "Some applications of two approximations to the multinomial distribution," *Biometrika*, vol. 47, pp. 463–469, 1960.

[18] Kolchin, V., Sevast'yanov, B., and Chistyakov, V., *Random Alloacations*. Washington, D.C.: V.H. Winston & Sons, 1978.

[19] Kozelka, R., "Approximate upper percentage points for extreme values in multinomial sampling," *Ann.Math.Statist.*, vol. 27, pp. 507–512, 1956.

[20] Leadbetter, M., Lindgren, G., and Rootzén, H., *Extremes and Related Properties of Random Sequences and Processes*. Springer-Verlag New York Inc., 1983.

[21] Levin, B., "On calculations involving the maximum cell frequency," *Comm. Statist.*, vol. 12, pp. 1299–1327, 1983.

[22] Magurran, A., *Measuring Biological Diversity*. Blackwell Publishing company, 2004.

[23] Mallows, C., "An inequality involving multinomial probabilities," *Biometrika*, vol. 55, pp. 422–424, 1968.

[24] May, R., "Patterns of species abundance and diversity," *Ecology and Evolution of communities*, 1975.

[25] May, R., "How many species?," *Phil. Trans. R. Soc. Lond. B*, 1990.

[26] May, R., "How many species inhabit on earth?," *Sci. Am*, 1992.

[27] May, R., "Conceptual aspects of the quantification of the exent of biological diversity," *Phil. Trans. R. Soc. Lond. B*, 1994.

[28] Molinary, J., "A critique of bulla's paper on diversity indices," *Oikos*, 1996.

[29] Olkin, I. and Sobel, M., "Integral expressions for tail probabilities of the multinomial and negative multinomial distributions," *Biometrika*, vol. 52, pp. 167–179, 1965.

[30] Patil, G. and Taillie, C., "Diversity as a concept and its measurement," *J.Amer.Statist. Assoc*, vol. 77, no. 379, pp. 548–561, 1982.

[31] Riordan, J., *An Introduction to Combinatorial Analysis*. NewYork: John Wiley & Sons, 1958.

[32] Rosenzweig, M., *Species diversity in space and time.* Cambridge, UK: Cambridge University Press, 1995.

[33] Rukhin, A., "Gamma distribution order statistics, maximal multinomial frequency and randomization designs," *JSPI*, vol. 136, pp. 2213–2226, 2006.

[34] Sobel, M., Uppuluri, V., and Frankowski, K., *Selected Tables in Mathematical Statistics*, vol. 4 of *Dirichlet Distribution-Type I. (IMS ed.).* Amer.Math.Soc, 1977.

[35] Tracy, C. and Widom, H., "Level-spacing distributions and the airy kernel," *Comm. Math. Phys.*, vol. 159, no. 1, 1994.

[36] Vershynin, R., *Introduction to the non-asymptotic analysis of random matrices, in Compressed Sensing, Theory and Applications.* ed. Y. Eldar and G. Kutyniok. Cambridge University Press., 2012.

[37] Weiss, L., "The normal approximation to the multinomial with an increasing number of classes," *Naval Research Logistics Quarterly*, vol. 23, pp. 139–149, 1976.

[38] Young, D., "Two alternatives to the standard χ^2 test of the hypothesis of equal cell frequencies," *Biometrika*, vol. 49, pp. 107–116, 1962.

VITA

Huy Huynh was born and raised in Da Nang, Vietnam. In August 2001, at seventeen, he came to the U.S. as an exchange student, living with an American family in Ada, Oklahoma. He graduated from Byng High School in May 2002. Afterwards, Huy moved to Atlanta and earned an Associate of Science degree in Computer Science at Georgia Perimeter College. Huy then transferred to Kennesaw State University and completed a Bachelor degree in Mathematics and was recognized there as Outstanding Math Senior. In August 2006, Huy enrolled at Georgia Institute of Technology to pursue a Doctorate in Mathematics. Since then, he has earned a Masters in Statistics from the School of Industrial and System Engineering. During his time at Georgia Tech, Huy has taught various courses both as an instructor and a teaching assistant. In 2010 he was awarded Georgia Tech's "CETL/BP Outstanding Graduate Teaching Assistant Award." In addition, he served as the Graduate Student Representative of the School of Mathematics in the academic years 2009-2010 and 2010-2011. Outside of the School of Mathematics, Huy has been involved in the Georgia Tech Student Foundations Investment Committee as a Quant Analyst and a Sector Analyst. Graduating in summer 2012, Huy will join the Department of Applied and Computational Mathematics and Statistics at the University of Notre Dame as a faculty member. In his spare time, Huy enjoys singing karaoke and playing badminton and table tennis with his family and friends.