Fall 2009

Bias in Closed Population Capture-Recapture

Candace M. Lynn

BIAS IN CLOSED POPULATION CAPTURE-RECAPTURE

by

CANDACE M. LYNN

(Under the Direction of Patricia Humphrey)

ABSTRACT

Our primary question was the effect of departures from assumptions on population estimates obtained using three maximum likelihood population size estimators: $\mathbf{M}_0$ (all individuals have the same capture probability on all occasions), $\mathbf{M}_b$ (behavior is affected by prior capture) and $\mathbf{M}_t$ (all individuals have the same capture probability that varies by occasion). After examining the initial results and observing substantial negative bias (underestimates), we attempted to model the bias for $\mathbf{M}_0$ and $\mathbf{M}_b$ capture scenarios as these situations had consistent patterns. The $\mathbf{M}_t$ scenario with its erratic behaviors was not modeled. We noted that $\mathbf{M}_0$ and $\mathbf{M}_t$ performed equally well for the $\mathbf{M}_0$ and $\mathbf{M}_b$ captures. $\mathbf{M}_b$ did better as an estimator for the $\mathbf{M}_b$ capture scenario than for the $\mathbf{M}_0$ scenario. The $\mathbf{M}_t$ estimator for $\mathbf{M}_t$ captures did not perform well. Depending on actual capture probabilities, either of the other two estimators may give better, less biased results.

INDEX WORDS: Statistical Simulation, Capture-Recapture, Maximum Likelihood, Closed Population, Bias Modeling

BIAS IN CLOSED POPULATION CAPTURE-RECAPTURE

by

CANDACE M. LYNN

B.S., Georgia Southern University, 2002

A Thesis Submitted to the Graduate Faculty of Georgia Southern University in Partial

Fulfillment

of the Requirements for the Degree

MASTER OF SCIENCE

STATESBORO, GEORGIA

2009

BIAS IN CLOSED POPULATION CAPTURE-RECAPTURE

by

CANDACE M. LYNN

Major Professor: Patricia Humphrey
Committee: Anna Oganyan
Charles Champ

Electronic Version Approved:
December 2009

ACKNOWLEDGMENTS

I thank my parents for their love, support, and encouragement while writing this thesis and throughout my entire education. I also thank my brother for his support. I have many other family members and friends who have given me encouragement and support during my graduate studies, and I thank them also.

I also thank Dr. Patricia Humphrey for her advisement and her encouragement in my graduate studies. I appreciate her sharing her time and knowledge with me while working on this thesis. I also thank my committee members for their willingness to serve.

I thank the faculty in the Mathematics Department at Georgia Southern University for their willingness to teach and share their knowledge with others. I especially thank those who taught me during my studies at Georgia Southern University.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

Page

CHAPTER 1

INTRODUCTION

Scientists are involved in many studies pertaining to animals, one of which is the estimation of population sizes to monitor species abundance, health of ecosystems, etc. One of the methods that is used is mark-recapture or capture-recapture. This method has been used for many years and several models have been developed. "The earliest forms of the capture-recapture method were used by LaPlace in 1802 to estimate the human population size of France and by John Graunt to estimate the effects of the plague and the size of the population in England in the early 1600s" (Amstrup). The models that were developed later are for both closed and open populations. A closed population is one in which the total number of individuals is not changing through births, deaths, immigration, or emigration. An open population is one that is (or could be) changing during the course of a study, because of any combination of births, deaths, immigration, or emigration. Several of the general models and some models that have recently been developed will be discussed in this chapter.

Several models have been created for closed populations. Each model that is created has assumptions that should be met. Some common assumptions for all population models are:

1. Animals do not lose their marks or tags.
2. All marks or tags are correctly recorded.
3. Animals act independently.

For a closed population model an additional assumption is that the population remains constant over the study period (e.g., there is no immigration or emigration), although known removals (e.g., deaths on capture) are allowed.

Lincoln and Petersen were two of the earliest scientists to apply this method to ecology. Petersen, along with Dahl, used the application on fish. They recognized that it was possible to model the fish population by marking the fish that were caught, and looking at the proportion of marked fish that were captured again. Lincoln also used this method to estimate duck populations by looking at the bands that had been placed on the ducks that were captured and then released. The Lincoln-Petersen method, which is easy to understand, has been used for closed populations. For this method, an initial sample, n_1, of the animal population is taken, marked and released. The marked portion of the population is

$$\frac{n_1}{N}$$

Another sample, n_2, is taken later, and the number of animals that are marked, m_2, is recorded. The proportion of marked animals in the second sample is m_2/n_2. Assuming that the marked proportion in the sample is equal to the marked proportion in the population suggests that

$$\frac{m_2}{n_2} \approx \frac{n_1}{N}$$

From this an estimate of the population size, which is known as the Lincoln-Petersen estimator is given by

$$\hat{N}_P = \frac{n_1 n_2}{m_2}$$

The Lincoln-Petersen method, although simple, is subject to bias if m_2 is small or zero, and it can overestimate the population. Therefore this method was modified by Chapman in 1951 to account for the bias that is possible. The estimator given by Chapman is

$$\hat{N}_C = \frac{(n_1+1)(n_2+1)}{m_2+1} - 1$$

Both the Lincoln-Petersen estimator and the Chapman estimator are for two sample models. While these are simple, the estimates of the populations are not very accurate. There was a need to have other variables involved in the estimation of population sizes and for the sampling to take place more than twice. "Schnabel in 1938 and Darroch in 1958 began to extend the closed population models" (Amstrup).

The models developed as an extension of the Lincoln-Peterson model are referred to as model $\mathbf{M}_0$ and model $\mathbf{M}_t$. "Both of these models have the assumption of equal catchability, *i.e.* each animal has the same probability of being captured each time sampling takes place. In model $\mathbf{M}_0$, the 0 refers to no variation in the capture probability, and likewise in model $\mathbf{M}_t$, the t refers to variation of the capture probabilities over time" (Amstrup). These models, however, are also not very accurate because of their assumption of equal catchability. This assumption is very easily violated due to the behavior of animals. Depending on the method of capture, animals may be either "trap happy" or "trap shy". The animal may like the bait that is being used to capture it and the same animal may return to the trap on many occasions to get the bait. This means the animal is "trap happy". The other case, "trap shy", occurs when the animal that was captured is unlikely to return to the trap again. Both cases reduce the accuracy of the model: if the animals are "trap happy", the ones that have been marked will return to the trap frequently therefore lowering the chance to trap unmarked animals and resulting in an estimate that is too low, whereas, if the animals are “trap shy”, the probability decreases of capturing the animals that are already marked, which results in an overestimation of population size. Heterogeneity is also a problem, i.e. one sex of the animal may be more catchable than the other, and the size of the animal can also play a role. These problems with the equal catchability assumption can lead to bias in estimating the population size when using these models in field work. The populations could be either underestimated or overestimated, depending on the reason for the violation. There are some solutions to these violations.

Modern closed population models take into account that animals do not have the same capture probability. Two types of this model are discrete-time and continuous-time models. In the discrete time model animals are caught on each occasion, but the exact time that they are caught is not recorded; therefore the order of the capture is unknown. “Otis et al. (1978) considered three sources of variations in capture probabilities: time effects, behavioral response to capture, and individual heterogeneity due to observable factors. Based on these sources of variation, Otis et al. (1978) and White et al. (1982) considered all possible combinations of sources, and formulated eight models” (Amstup). To describe the models let P_{ij} denote the probability the ith individual is captured on the jth occasion. The subscripts t, b, h, on **M** denote time variation, behavioral response, and heterogeneity, respectively. The subscript 0 denotes the null model. The models are as follows:

Model $\mathbf{M}_0$: $P_{ij} = p$

Model $\mathbf{M}_t$: $P_{ij} = p_j$

Model $\mathbf{M}_b$: $P_{ij} = p$ until the first capture and $P_{ij} = c$ for any recaptures

Model $\mathbf{M}_{tb}$: $P_{ij} = p_j$ until the first capture and $P_{ij} = c_j$ for any recaptures

Model $\mathbf{M}_{h}$: $P_{ij} = p_i$

Model $\mathbf{M}_{bh}$: $P_{ij} = p_i$ until first capture and $P_{ij} = c_i$ for any recapture

Model $\mathbf{M}_{th}$: $P_{ij} = p_i e_j$

Model $\mathbf{M}_{tbh}$: $P_{ij} = p_{ij}$ until first capture and $P_{ij} = c_{ij}$ for any recapture

Each model has weaknesses; however some of these models have been identified as being useful for estimating a specific type of animal. Model $\mathbf{M}_{tb}$ has been used for some quail and mice populations. "Model $\mathbf{M}_{h}$ is useful for many species such as rabbits, chipmunks, skunks, and grizzly bears. Model $\mathbf{M}_{bh}$ has been useful in fisheries, and Model $\mathbf{M}_{tbh}$ has been selected as the most likely model for estimating the size of squirrel and mouse populations" (Amstrup). Some new approaches to these models have been explored using numerical computation. These include using bootstrap methods, improved interval estimation, maximum likelihood estimation for Model $\mathbf{M}_{tb}$, the jackknife technique for Model $\mathbf{M}_{bh}$, sample coverage approaches for Models $\mathbf{M}_{h}$ and $\mathbf{M}_{th}$, and estimating equations (including maximum quasi-likelihood and martingale methods by Yip et al. which have simpler estimating equations that in some cases become equivalent to the classical). Some other methods that have also been explored are log-linear or generalized linear models, Bayesian methods, parametric approaches for modeling heterogeneity, latent class, mixture model, and nonparametric maximum likelihood.

The continuous-time model has one capture in each capture occasion and has the exact time of each capture recorded. This can be useful with large animals such as whales. "As in the discrete-time models, a series of eight continuous-time models can be postulated depending on the sources of variability in the Poisson rates due to time, behavioral response, and heterogeneity. Consider a Poisson process with parameter $\lambda_i^*(t)$, which can be intuitively interpreted as capturing animal *i* in a small time interval around time *t*. The Poisson rate for model $\mathbf{M}_{tbh}$ is given by

$$\lambda_i^*(t) = \begin{cases} \lambda_i \alpha(t) & \text{until first capture} \\ \phi \lambda_i \alpha(t) & \text{for any recapture} \end{cases} .$$

Here $\alpha(t)$, $\{\lambda_1, \lambda_2, ..., \lambda_N\}$, and ϕ represent the effects of time, heterogeneity, and behavioral response, and $\alpha(t)$ is an arbitrary time-varying function in (0, *T*)" (Amstrup).

The eight continuous models are given below:

Model $\mathbf{M}_{bh}$: set $\alpha(t) = 1$ in model $\mathbf{M}_{tbh}$

Model $\mathbf{M}_{tb}$: set $\lambda_i = 1$ in model $\mathbf{M}_{tbh}$

Model $\mathbf{M}_{th}$: set $\phi = 1$ in model $\mathbf{M}_{tbh}$

Model $\mathbf{M}_{h}$: set $\alpha(t) = 1$ and $\phi = 1$ in model $\mathbf{M}_{tbh}$

Model $\mathbf{M}_{b}$: $\alpha(t) = 1$, $\lambda_i = \lambda$ in model $\mathbf{M}_{tbh}$

Model $\mathbf{M}_{t}$: set $\lambda_i = 1$ and $\phi = 1$ in model $\mathbf{M}_{tbh}$

Model $\mathbf{M}_{0}$: set $\alpha(t) = 1$, $\lambda_i = \lambda$ and $\phi = 1$ in model $\mathbf{M}_{tbh}$.

There has been little published on these models.

A classic model for open populations is the Jolly-Seber model which was first published in 1965. This model was influenced in development by papers published by Darroch in 1959 and Cormack in 1964. "Darroch presented maximum likelihood estimators for the additions-only model and the deletions-only model. Cormack considered survival and capture probability estimation for marked birds and derived one component of the likelihood used by Jolly and Seber in their more general model" (Amstrup). For the Jolly-Seber model, there are several assumptions:

1. every animal alive in the population at a given sample time j has an equal chance (p_j) of being captured in that sample (equal catchability);
2. every marked animal alive in the population at a given sample time j has an equal chance of survival (ϕ_j) until the next sampling occasion (implicitly this assumption applies to all animals, marked and unmarked, in order to estimate the survival of all animals in the population);
3. marked animals do not lose their marks and marks are not overlooked;
4. sampling periods are short (i.e. effectively instantaneous); and
5. all emigration from the population is permanent.

The Jolly-Seber model uses the maximum likelihood approach to obtain parameter estimates for the population size, survival probability, capture probability, the number of marked animals in the population, and the total number of new animals entering the population. There may be some problems with the estimates of population size and survival rate due to the animals being "trap shy" or "trap happy" and heterogeneity. Problems may also exist from tag loss or from problems with the marking of animals, e.g. tag-induced mortality. These can cause the results to be either positively or negatively biased. In addition to practical problems, the method can get improper estimates, e.g. negative birth rates, survival probability greater than one, and some desired parameters are not estimable. Newer models have been explored to look at these issues for both the closed and open populations.

For modern open population models two classes exist. The first is conditional and the second is unconditional. In the conditional case, there are single-age models. The conditional Cormack-Jolly-Seber (CJS) model is in this category. With increased use of computers, additional models have been developed which can look at elements that could not have been easily considered earlier. These include the reduced parameter model, time-specific covariates, multiple groups, effects of capture history, and individual covariates. There are also multiple-age models such as Pollock's multiple-age model, age 0 cohort models, and age-specific breeding models. Reverse-time models have also been used to look at the recruitment process. The Jolly-Seber (JS) model is an unconditional model. A superpopulation approach and a temporal symmetry approach have also been developed.

This thesis will focus on an examination of model performance of closed populations under various conditions that possibly violate model assumptions. Different models have different requirements for practical field work such as individual tags and capture histories versus batch marks which make field work easier. The biologist (or demographer) must make an educated guess about which model to use in a given situation. We hope to determine if one or more models are generally adequate (robust) in the sense of providing reasonable estimates with greater ease-of-use. Little or no work has been done in this area to date. The only paper found on the topic was by Manly, et.al. and pertained to the population of the northern spotted owl (1999).

CHAPTER 2

MATHEMATICAL MODELS FOR CLOSED POPULATION CAPTURE-RECAPTURE

There are many different methods that can be used to obtain population parameter estimates. For the capture-recapture method of obtaining population estimates for closed populations under certain assumptions, one of the methods that is used is the method of maximum likelihood estimation. The likelihood function is a function which tells us the likelihood of the population parameter(s), given the observed data. The maximum likelihood estimate(s) is the most likely value of the parameter(s) given what is observed. The method of maximum likelihood consists of two steps, the first of which is the construction of a model that states the probability of observing the data as a function of the unknown parameters that are of interest. In the second step, the estimates of the unknown parameters are chosen to be those values that make the likelihood function from the first step as large as possible, i.e., the values that maximize the likelihood. For models $\mathbf{M}_0$, $\mathbf{M}_b$, and $\mathbf{M}_t$ in closed populations this method can be used numerically to estimate the value of the total population. The maximum likelihood estimators (MLEs) of $\hat{N}$ and $\hat{p}$ are those values of N and p which maximize the function

$$L(N, p) = P\{X_{ij} \mid N, p\}$$

or equivalently, which maximize the log of $L(N,p)$. The derivation of these estimators is described in detail by Otis, et. al. (1978). The following are the statistics and notation that is used:

Section 2.1 Notation

$$X_{ij} = \begin{cases} 1 \text{ individual i captured on occasion j} \\ 0 \text{ otherwise} \end{cases}$$

N = population size

n_j = the number of animals captured in the j^{th} sample, $j = 1,2,\ldots,t$, $= \sum_{t=1}^{N} X_{ij}$

$n.$ = the total number of captures during the study $= \sum_{j=1}^{t} n_j$

u_j = the number of new (unmarked) animals captured in the j^{th} sample, $j = 1,2,\ldots,t$

M_{t+1} = the number of distinct individuals caught during the experiment (recall that t is fixed for a given experiment) $= \sum_{j=1}^{t} u_j$

M_j = the number of marked animals in the population at the time of the j^{th} sample, $j = 2,3,\ldots,t$. (Note that $M_1 \equiv 0$)

$M.$ = sum of the M_j [does not include M_{t+1}] $= \sum_{j=1}^{t} M_j$

m_j = the number of marked animals captured in the j^{th} sample, $j = 2,\ldots,t$. Note that $u_j = n_j - m_j$ and that $m_1 = 0$

$m.$ = sum of the m_j $= \sum_{j=1}^{t} m_j$

t = number of capture occasions

p = probability of capture for any individual in $\mathbf{M}_0$, in $\mathbf{M}_b$ the probability of capture of an unmarked individual

c = probability of capture of a marked individual in model $\mathbf{M}_b$

Section 2.2 Maximum Likelihood Basics in Mark-Recapture

Model $\mathbf{M}_0$ is the simplest of the models for a closed population, because there is no variation in capture probability. This is comparable to the conditions of an urn experiment, where the urn contains a set of marbles of a given color, each of which has an equal probability of being picked. A sampling scheme occurs where marbles are randomly selected at each sampling occasion. At each occasion, the number of marbles that are “marked” and “unmarked” are recorded, and the marbles with the original color are replaced with another color before being returned to the urn. This process is repeated several times, and an estimation of the population (the total number of balls) is made based on these samples. However this idea gives rise to a hypergeometric distribution for which MLEs do not exist. Since we assume animals are independent of one another, this gives rise to a binomial sampling distribution (the hypergeometric does not have this assumption). The probability of capturing n_1 animals for the first capture occasion based on a binomial distribution is:

$$L(n_1) = \binom{N}{n_1} p^{n_1} (1-p)^{N-n_1}$$

For the second occasion, the likelihood conditional on the first occasion is

$$L(u_2, m_2 \mid M_2) = \binom{M_2}{m_2} p^{m_2} (1-p)^{M_2-m_2} \binom{N-M_2}{u_2} p^{u_2} (1-p)^{N-M_2-u_2}$$

$$= \binom{M_2}{m_2}\binom{N-M_2}{u_2} p^{n_2} (1-p)^{N-n_2}$$

If we multiply across all capture occasions for an overall unconditional likelihood for the entire study we get

$$L(x)=\prod_{s=1}^{t}\binom{U_s}{u_s}p^{u_s}(1-p)^{U_s-u_s}\binom{M_s}{m_s}p^{m_s}(1-p)^{M_s-m_s}$$

$$=\prod_{s=1}^{t}\binom{U_s}{u_s}\binom{M_s}{m_s}p^{n_s}(1-p)^{N_s-n_s}$$

The product of this cannot be maximized, so we take the log of the function.

Section 2.3 Model M_0 Likelihood

In Model $\mathbf{M}_0$, parameterized by the parameters N and p, the relevant part of the log-likelihood function is given by

$$\ln L(N,p\mid \boldsymbol{X})=\ln\left(\frac{N!}{(N-M_{t+1})!}\right)+(n.)\ln(p)+(tN-n.)\ln(1-p),$$

where $p\in[0,1]$ and $N\in\mathcal{N}=\{M_{t+1},M_{t+1}+1,M_{t+1}+2,...\}$. Given the value of N, the MLE $\hat{p}(N)$ of p is given as the solution to

$$\frac{\partial}{\partial p}\ln L(p\mid N,\boldsymbol{X})=0\,,$$

which reduces to

$$\frac{n.}{\hat{p}(N)}=\frac{tN-n.}{1-\hat{p}(N)}\ .$$

This results in the solution

$$\hat{p}(N)=\frac{n.}{tN}\ .$$

Now, the MLE $\hat{N}_o$ of N satisfies

$$\ln L(\hat{N}_o,\hat{p}(\hat{N}_o)\mid \mathrm{X})=\max_{N\in\mathcal{N}}\left[\max_{p\in[0,1]}\ln L(p\mid N,\mathrm{X})\right]$$

$$= \max_{N \in \mathcal{N}} \left[\ln L(\hat{p}(N) \mid N, \boldsymbol{X}) \right]$$

$$= \max_{N \in \mathcal{N}} \left[\begin{array}{l} \ln\left(\dfrac{N!}{(N - M_{t+1})!} \right) + (n.)\ln(n.) + \\ (tN - n.)\ln(tN - n.) - tN\ln(tN) \end{array} \right].$$

For a given data set, a search over $\mathcal{N}$ is performed to locate the MLE $\hat{N}_o$. This value is then used in the calculation of the MLE $\hat{p}(\hat{N}_o) = \hat{p}$ via

$$\hat{p} = \frac{n.}{t\hat{N}_o}.$$

The asymptotic variance of $\hat{N}_o$ was derived by Darroch 1959 as

$$Var(\hat{N}_o) = N[(1-p)^{-t} - t(1-p)^{-1} + t - 1]^{-1} \; .$$

An estimate of this variance is

$$\hat{V}ar(\hat{N}_o) = \hat{N}_o[(1-\hat{p})^{-t} - t(1-\hat{p})^{-1} + t - 1]^{-1} \; .$$

Section 2.4 Model $\mathbf{M_b}$ Likelihood

Model $\mathbf{M}_b$ is parameterized by the parameters *N*, *p*, and *c*. The likelihood function can be constructed similarly to that in Section 2.2, but using *c* for the capture probability of marked individuals. The part of the log-likelihood necessary for estimation of the parameters is given by

$$\begin{array}{l} \ln L(N, p, c \mid \boldsymbol{X}) = \ln\left(\dfrac{N!}{(N - M_{t+1})!} \right) + M_{t+1}\ln(p) + (tN - M. - M_{t+1})\ln(1-p) \\ \qquad + m.\ln(c) + (M. - m.)\ln(1-c) \end{array},$$

where $N \in \mathcal{N} = \{M_{t+1}, M_{t+1}+1, M_{t+1}+2, \ldots\}$, $p \in [0,1]$, $c \in [0,1]$. The MLE $\hat{c}$ of c is produced by the equation

$$\frac{\partial}{\partial c} \ln L(N, p, c \mid \boldsymbol{X}) = 0,$$

which reduces to

$$\frac{m.}{\hat{c}} = \frac{M. - m.}{1-\hat{c}}.$$

Solving this equation gives $\hat{c} = m./M.$. Thus, we see that estimation of c is independent of the estimation of N and p. Now, the relevant part of the log-likelihood function for purposes of estimating N and p is

$$\ln L(N, p \mid \boldsymbol{X}) = \ln\left(\frac{N!}{(N-M_{t+1})!}\right) + M_{t+1}\ln(p) + (tN - M. - M_{t+1})\ln(1-p).$$

Given the value of N, the MLE $\hat{p}(N)$ of p is provided by the equation

$$\frac{\partial}{\partial p} \ln L(p \mid N, \boldsymbol{X}) = 0,$$

which reduces to

$$\frac{M_{t+1}}{\hat{p}(N)} = \frac{tN - M. - M_{t+1}}{1-\hat{p}(N)}.$$

The solution to this equation gives

$$\hat{p}(N) = \frac{M_{t+1}}{tN - M.}$$

Now, the MLE $\hat{N}_b$ of N satisfies

$$\ln L(\hat{N}_b, \hat{p}(\hat{N}_b) \mid \boldsymbol{X}) = \max_{N \in \mathcal{N}} \left[\max_{p \in [0,1]} \ln L(p \mid N, \boldsymbol{X})\right]$$

$$= \max_{N \in \mathcal{N}} \left[\ln L(\hat{p}(N) \mid N, \boldsymbol{X})\right]$$

$$= \max_{N \in \mathcal{N}} \left[\begin{array}{l} \ln\left(\dfrac{N!}{(N - M_{t+1})!}\right) + M_{t+1}\ln(M_{t+1}) + \\ (tN - M. - M_{t+1})\ln(tN - M. - M_{t+1}) \\ -(tN - M.)\ln(tN - M.) \\ \end{array} \right].$$

A search over $\mathcal{N}$ is performed to locate the MLE $\hat{N}_b$. It is then possible to calculate the MLE $\hat{p}(\hat{N}_b) = \hat{p}$ of p via

$$\hat{p} = \frac{M_{t+1}}{t\hat{N}_b - M.}.$$

The asymptotic variance of $\hat{N}_b$ is given by Zippin (1956) as:

$$Var(\hat{N}_b) = \frac{N(1-p)^t[1-(1-p)^t]}{[1-(1-p)^t]^2 - t^2 p^2 (1-p)^{t-1}} .$$

An estimate of this variance is given by

$$\hat{Var}(\hat{N}_b) = \frac{\hat{N}_b(1-\hat{p})^t[1-(1-\hat{p})^t]}{[1-(1-\hat{p})^t]^2 - t^2 \hat{p}^2 (1-\hat{p})^{t-1}} .$$

Section 2.5 Model $\mathbf{M_t}$ Likelihood

Model $\mathbf{M}_t$ is parameterized by the $t + 1$ parameters N, $p_1, p_2, ..., p_t$. Proceeding as discussed previously to construct the likelihood, the relevant log-likelihood function for estimation of the parameters is given by

$$\ln L(N, \boldsymbol{p} \mid \boldsymbol{X}) = \ln\left(\frac{N!}{(N - M_{t+1})!}\right) + \sum_{j=1}^{t} n_j \ln(p_j) + \sum_{j=1}^{t} (N - n_j)\ln(1 - p_j),$$

where $N \in \mathcal{N} = \{ M_{t+1}, M_{t+1}+1, M_{t+1}+2, ... \}$, $\boldsymbol{p} = \{ p_1, p_2, ..., p_t \}$, $p_j \in [0,1]$ for j = *1,2,…,t*. Given the value of *N*, the MLEs $\hat{p}_j(N)$ of p_j are given as the solutions to the system of equations

$$\frac{\partial}{\partial p_j} \ln L(\boldsymbol{p} \mid N, \boldsymbol{X}) = 0, \quad j = 1,2,\ldots,t.$$

The j^{th} one of these equations reduces to

$$\frac{n_j}{\hat{p}_j(N)} = \frac{N - n_j}{1 - \hat{p}_j(N)},$$

which results in the solution

$$\hat{p}_j(N) = \frac{n_j}{N}.$$

Now, the MLE $\hat{N}_t$ of *N* satisfies

$$\begin{aligned}
\ln L(\hat{N}_t, \hat{p}_1(\hat{N}_t), ..., \hat{p}_t(\hat{N}_t) \mid \boldsymbol{X}) &= \max_{N \in \mathcal{N}} \left[\max_{p_j \in [0,1]} \ln L(p_1, p_2, ..., p_t \mid N, \boldsymbol{X}) \right] \\
&= \max_{N \in \mathcal{N}} \left[\ln L(\hat{p}_1(N), \hat{p}_2(N), ..., \hat{p}_t(N) \mid N, \boldsymbol{X}) \right] \\
&= \max_{N \in \mathcal{N}} \left[\begin{array}{l} \ln\left(\dfrac{N!}{(N - M_{t+1})!} \right) \\ + \sum_{j=1}^{t} n_j \ln(n_j) + \sum_{j=1}^{t} (N - n_j) \ln(N - n_j) - tN \ln(N) \end{array} \right].
\end{aligned}$$

A search over $\mathcal{N}$ is performed in order to locate the MLE $\hat{N}_t$. It is then possible to calculate the MLEs $\hat{p}_j(\hat{N}_t) = \hat{p}_j$ of the p_j for *j = 1,2,…,t* via

$$\hat{p}_j = \frac{n_j}{\hat{N}_t} .$$

The asymptotic variance of $\hat{N}_t$ is given by Darroch (1958) as:

$$Var(\hat{N}_t) = N\left[\frac{1}{\prod_{j=1}^{t}(1-p_j)} + t - 1 - \sum_{j=1}^{t}(1-p_j)^{-1}\right]^{-1} .$$

An estimate of this variance is

$$\hat{V}ar(\hat{N}_t) = \hat{N}_t\left[\frac{1}{\prod_{j=1}^{t}(1-\hat{p}_j)} + t - 1 - \sum_{j=1}^{t}(1-\hat{p}_j)^{-1}\right]^{-1} .$$

Section 2.6 Behavior of the Estimates

It has been previously noted and proven that estimates of the population size are asymptotically unbiased (as $t \to \infty$), because (intuitively) if enough occasions, all individuals will eventually be captured. Of more interest is behavior for a limited number of occasions – what is the bias (under/over estimate) if any? It is commonly known in statistical circles that many MLEs are biased. How do these models perform when the underlying assumptions are violated? The latter question is the primary focus of the next chapter, and will be studied via simulation.

CHAPTER 3

THE HOMOGENEOUS CAPTURE SCENARIO - BIAS OF ESTIMATORS

Section 3.1 Model M_0 Simulation Program

A program was written in Fortran to simulate a closed population with equal catchability using the binomial capture model. All of the individuals have the same capture probability on all occasions and behave independently in terms of captures. This program allowed us to specify the population size N and the number of repetitions of the simulation. The capture probability, p, was varied from 0.1 to 0.8 and the number of capture occasions was varied from 3 to 10 to be able to examine the effects of the study parameters. These capture probabilities were selected based on estimated probabilities found in the literature where recapture probabilities have been estimated as high as 88 percent. From the simulated capture data, maximum likelihood estimates of the population size and their standard errors were computed for estimators $\mathbf{M}_0$, $\mathbf{M}_b$ and $\mathbf{M}_t$, as shown in Table 3.1. If all three models are equivalent they should result in similar estimates of the population. To analyze the simulated results, graphics were generated to look for patterns.

Figure 3.1 shows the effects of increased numbers of capture occasions for given values of p. We clearly see that the average estimate of population size increases with an increase in the number of capture occasions. We can also see a clear negative bias in the average estimates with smaller capture probabilities. In addition, model $\mathbf{M}_b$ tends to give smaller estimates than the other models until capture probabilities are sufficiently large. This behavior is most likely due to the fact that $\mathbf{M}_b$ only uses information on new captures for population estimation. $\mathbf{M}_0$ and $\mathbf{M}_t$ give essentially the same results. For this capture scenario, models $\mathbf{M}_0$ and $\mathbf{M}_t$ should be equivalent; the only difference between the two being that $\mathbf{M}_t$ allows capture probabilities to vary by occasion.

Table 3.1 Model M_0 Simulation Results

nocc	p	Nhat M0	SE_M0	Nhat Mb	SE_Mb	Nhat_Mt	SE_Mt
3	0.1	304.536	366.151	285.504	199.440	304.424	365.584
4	0.1	386.428	424.892	362.252	238.290	386.324	424.433
5	0.1	459.516	447.211	432.018	263.111	459.430	446.881
6	0.1	523.010	449.904	493.422	275.293	522.934	449.647
7	0.1	578.438	438.832	547.948	282.713	578.348	438.566
8	0.1	628.040	420.928	597.340	283.617	627.940	420.668
9	0.1	669.848	393.654	639.790	276.790	669.784	393.505
10	0.1	707.868	363.722	678.776	263.920	707.776	363.535
3	0.2	536.256	423.651	510.766	269.848	536.150	423.278
4	0.2	642.890	394.785	615.846	275.259	642.788	394.515
5	0.2	725.648	344.462	699.640	259.090	725.584	344.330
6	0.2	787.012	285.620	763.783	229.763	786.928	285.490
7	0.2	835.576	233.261	815.550	198.051	835.522	233.195
8	0.2	872.082	187.053	855.826	170.122	872.024	186.997
9	0.2	900.418	149.975	887.213	142.008	900.338	149.914
10	0.2	922.366	119.075	912.089	117.429	922.346	119.063
3	0.3	706.448	346.692	683.130	261.038	706.370	346.504
4	0.3	806.517	263.914	786.207	219.433	806.476	263.849
5	0.3	870.814	188.119	855.264	169.691	870.750	188.052
6	0.3	913.692	131.733	902.698	127.078	913.648	131.701
7	0.3	940.434	90.029	933.034	89.827	940.416	90.020
8	0.3	960.072	61.910	955.350	63.859	960.043	61.900
9	0.3	972.687	43.239	969.668	45.172	972.671	43.235
10	0.3	970.782	28.001	979.450	31.636	970.782	28.001
3	0.4	825.476	237.215	808.120	201.261	825.416	237.123
4	0.4	901.503	144.269	890.130	136.764	901.447	144.219
5	0.4	944.138	85.617	937.422	85.470	944.088	85.591
6	0.4	968.477	50.357	964.934	52.289	968.430	50.342
7	0.4	980.987	29.387	979.244	30.892	980.951	29.380
8	0.4	989.276	17.384	988.420	18.275	989.254	17.382
9	0.4	988.896	9.993	993.098	10.810	988.896	9.993
10	0.4	993.458	5.996	996.493	6.346	993.458	5.996
3	0.5	905.190	139.811	894.867	132.419	905.136	139.759
4	0.5	955.730	68.262	950.778	69.176	955.710	68.253
5	0.5	978.799	33.335	976.736	34.726	978.777	33.330
6	0.5	989.556	16.207	988.882	17.129	989.538	16.205
7	0.5	995.347	8.026	994.995	8.373	995.341	8.026
8	0.5	997.634	3.960	997.602	4.140	997.622	3.959

nocc	p	Nhat M0	SE_M0	Nhat Mb	SE_Mb	Nhat_Mt	SE_Mt
9	0.5	997.294	1.925	999.288	2.040	997.294	1.925
10	0.5	998.229	0.967	999.655	0.992	998.229	0.967
3	0.6	954.178	70.648	949.450	72.064	954.140	70.626
4	0.6	982.773	27.243	981.350	28.791	982.745	27.237
5	0.6	993.010	10.556	992.576	10.955	992.990	10.554
6	0.6	997.044	4.128	996.936	4.244	997.036	4.128
7	0.6	999.265	1.661	999.253	1.706	999.265	1.661
8	0.6	999.229	0.658	999.258	0.643	999.229	0.658
9	0.6	998.611	0.259	999.614	0.265	998.611	0.259
10	0.6	998.853	0.103	999.857	0.106	998.853	0.103
3	0.7	981.549	28.889	980.002	30.121	981.529	28.883
4	0.7	994.796	8.352	994.497	8.742	994.794	8.352
5	0.7	998.348	2.450	998.348	2.521	998.348	2.450
6	0.7	999.177	0.730	999.258	0.730	999.177	0.730
7	0.7	999.725	0.220	999.725	0.218	999.725	0.220
8	0.7	999.926	0.066	999.926	0.067	999.926	0.066
9	0.7	999.982	0.020	999.982	0.019	999.982	0.020
10	0.7	998.985	0.006	999.990	0.006	998.985	0.006
3	0.8	994.659	8.335	994.325	8.684	994.645	8.333
4	0.8	999.269	1.641	999.235	1.688	999.269	1.641
5	0.8	999.596	0.324	999.596	0.320	999.596	0.324
6	0.8	999.922	0.064	999.922	0.063	999.922	0.064
7	0.8	999.984	0.013	999.984	0.013	999.984	0.013
8	0.8	999.994	0.003	999.994	0.003	999.994	0.003
9	0.8	1000	0.001	1000	0.001	1000	0.001
10	0.8	1000	0	1000	0	1000	0

Figure 3.1 Model M_0 Capture Occasion Effect Given Capture Probability

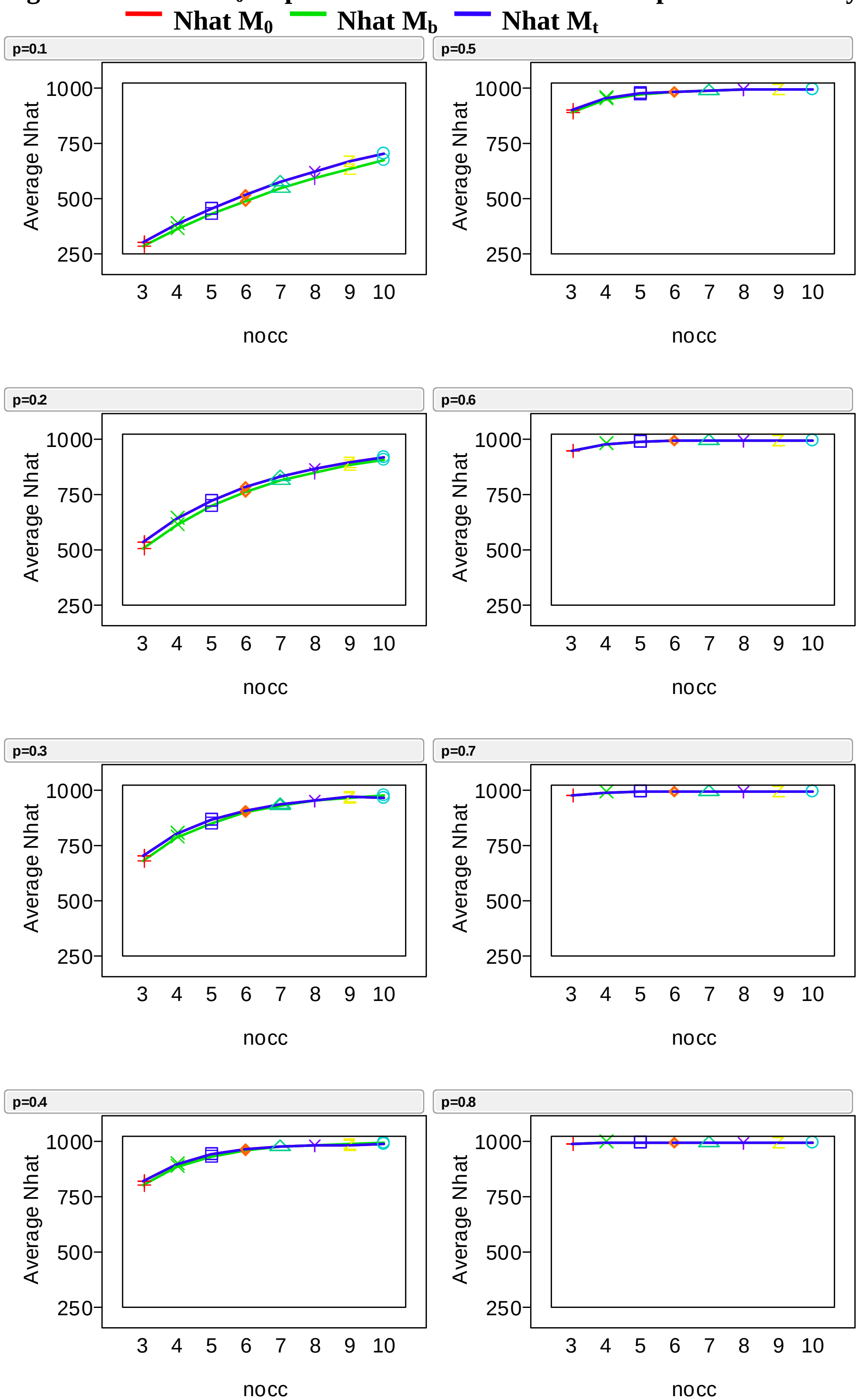


Figure 3.2 shows the effects of increased values of p for given values of the number of capture occasions (nocc). As p increased, the average estimates of the population size increased, and all the models gave essentially the same estimates. The $\mathbf{M}_b$ estimates of the population are slightly lower for the lower values of p for all nocc values, as seen previously. As nocc increased, the bias decreased, although there is still substantial negative bias for low values of p even with nocc large. Essentially what is seen is that as nocc increases and p increases, the estimated values become very close to the actual population value.

The standard errors for $\mathbf{M}_0$ and $\mathbf{M}_b$ were also computed in the Fortran simulation program for model $\mathbf{M}_0$. Figures 3.3 and 3.4 both show that as the number of capture occasions and the value of p increase, the standard errors of the estimates decrease. With all estimates biased low, we are interested in what combinations of capture probability and number of sampling occasions will get the estimated population size close to the actual size. As can be seen in the graphs, we will need at least four sampling occasions and a capture probability of at least 0.2 before this happens.

Figure 3.2 Model M_0 Capture Probability Effect Given Capture Occasions

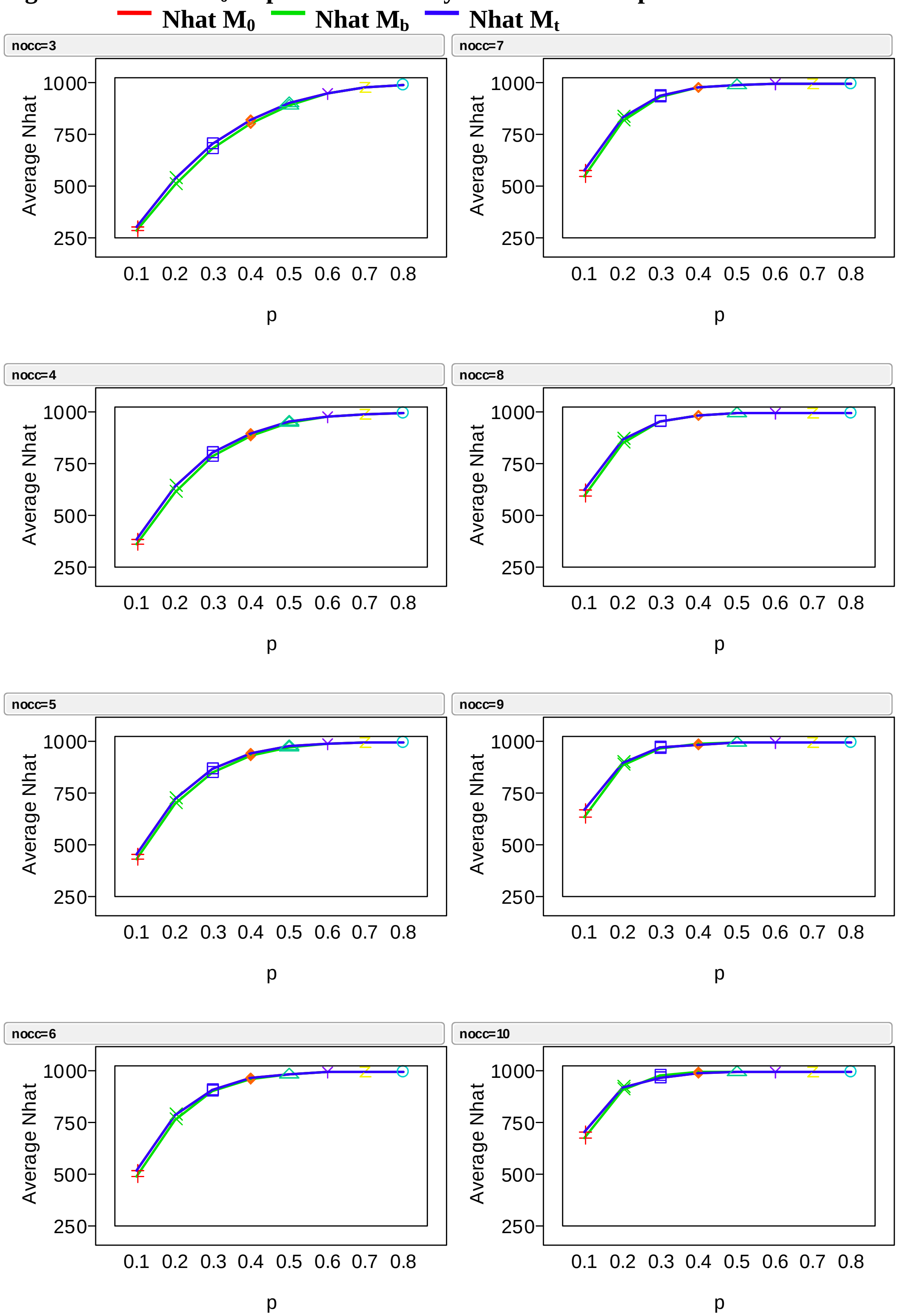


Figure 3.3 Estimated Value of M_0 With Standard Errors Added and Subtracted

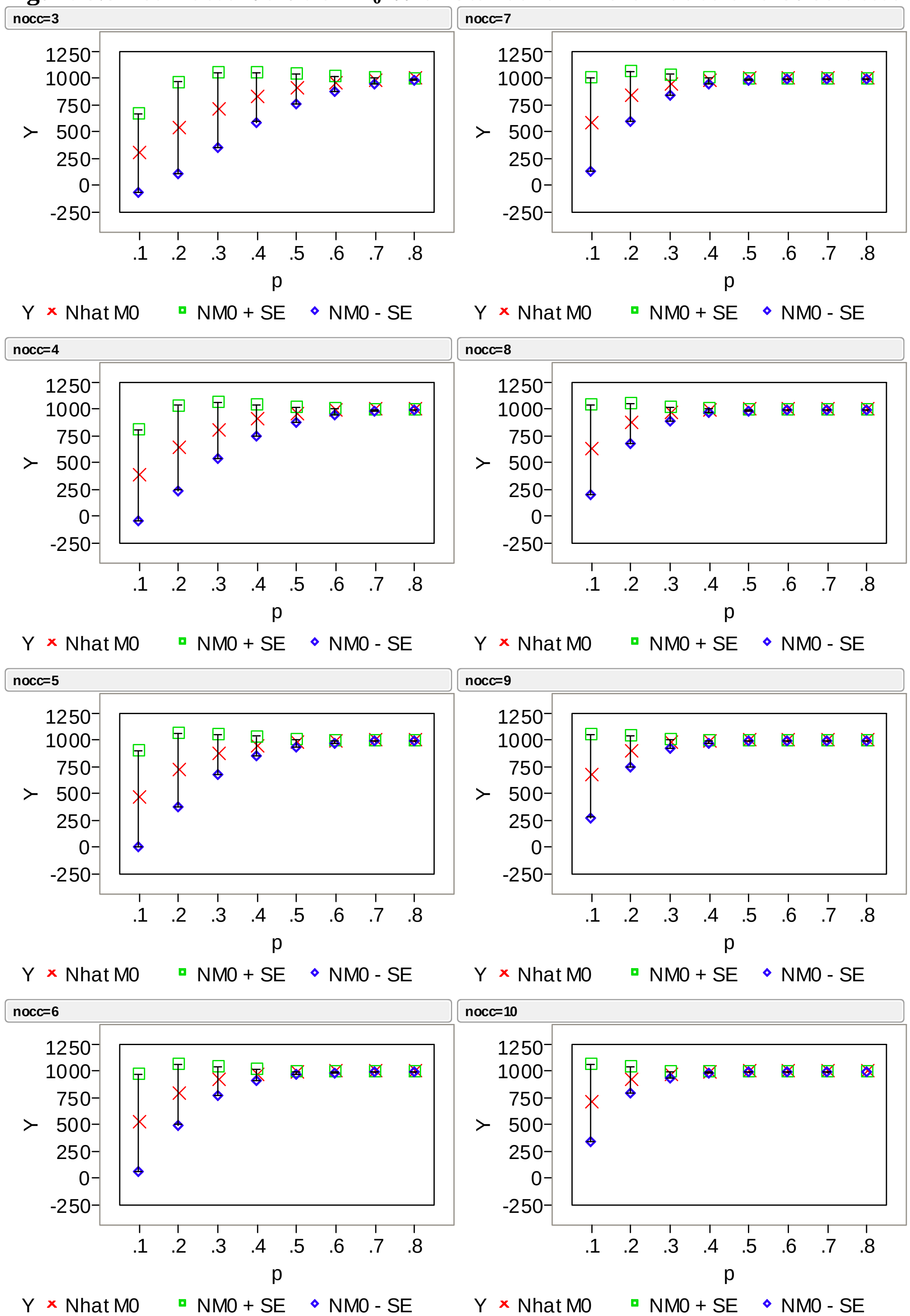
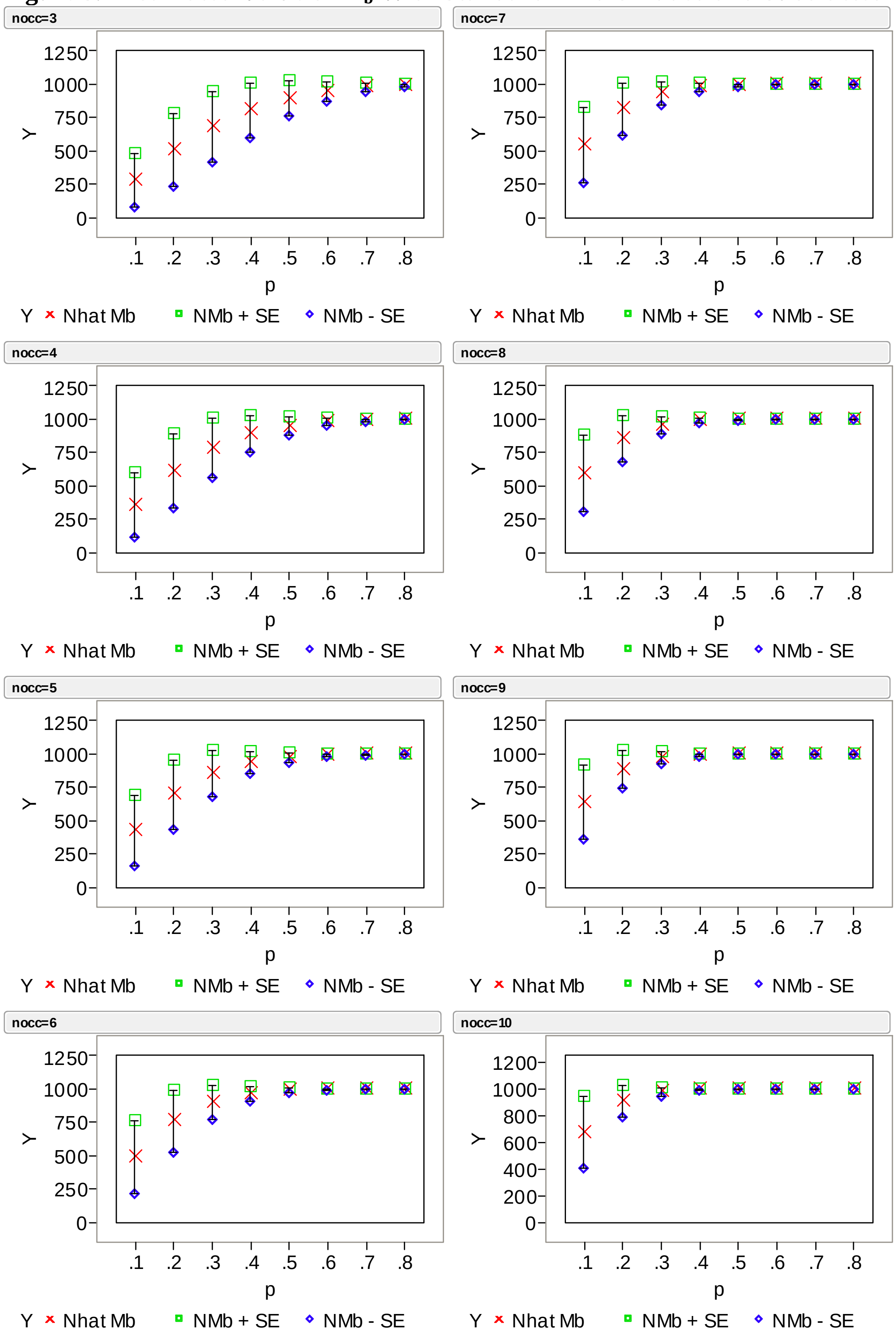


Figure 3.4 Estimated Value of M_b With Standard Errors Added and Subtracted



Section 3.2 Bias Modeling

We first reran the simulation for various values of N to determine whether the bias seen for N=1000 was consistent. At all population levels examined, the model consistently estimated the same proportion. To model the bias, defined as 100 minus the estimated percent, in terms of nocc and p, we noted that the bias was not a linear function of either nocc or p. We attempted to model this with a nonlinear model that had an asymptote at 0 (no bias) but were unsuccessful. In every model examined the hessian became singular so parameter estimates were unreliable. Figure 3.5 shows the results from a regression analysis which was performed in Minitab. These results were generated after the data set had been trimmed. We removed simulation cases after the MLE had essentially no bias (the MLE estimated at least 99 percent of the actual population) for a given number of captures, to reduce the weight these cases had on the regression. The values that were trimmed are listed in Table 3.2.

Table 3.2 Criteria for Data Trimming of Model M_0 Results

nocc	pcap greater than or equal to
4	0.8
5	0.7
6	0.7
7	0.6
8	0.6
9	0.6
10	0.5

Figure 3.5 Regression Analysis Results for M_0 bias in Model M_0

Regression Analysis: m0bias versus pcappct, pcapSq, nocc, nocc*pcap

The regression equation is
m0bias = 116 - 3.39 pcappct + 0.0238 pcapSq - 7.57 nocc + 0.133 nocc*pcap

46 cases used 18 cases contain missing values

Predictor	Coef	SE Coef	T	P
Constant	115.971	4.912	23.61	0.000
pcappct	-3.3863	0.2173	-15.59	0.000
pcapSq	0.023840	0.001943	12.27	0.000
nocc	-7.5680	0.6031	-12.55	0.000
nocc*pca	0.13272	0.01815	7.31	0.000

S = 3.849   R-Sq = 96.1%   R-Sq(adj) = 95.7%

Analysis of Variance

Source	DF	SS	MS	F	P
Regression	4	14789.6	3697.4	249.54	0.000
Residual Error	41	607.5	14.8		
Total	45	15397.1			

Source	DF	Seq SS
pcappct	1	8676.0
pcapSq	1	2646.3
nocc	1	2674.8
nocc*pca	1	792.4

Unusual Observations

Obs	pcappct	m0bias	Fit	SE Fit	Residual	St Resid
8	80.0	0.502	6.790	2.568	-6.288	-2.19RX
16	80.0	*	9.840	2.714	*	* X
24	80.0	*	12.889	3.154	*	* X
31	70.0	*	6.079	2.308	*	* X
32	80.0	*	15.939	3.787	*	* X
39	70.0	*	7.802	2.931	*	* X
40	80.0	*	18.989	4.533	*	* X
46	60.0	*	1.778	2.298	*	* X
47	70.0	*	9.524	3.616	*	* X
48	80.0	*	22.039	5.344	*	* X
54	60.0	*	2.173	2.865	*	* X
55	70.0	*	11.247	4.332	*	* X
56	80.0	*	25.088	6.196	*	* X
57	10.0	29.313	22.084	1.892	7.229	2.16R
61	50.0	*	-3.064	2.222	*	* X
62	60.0	*	2.569	3.447	*	* X
63	70.0	*	12.969	5.067	*	* X
64	80.0	*	28.138	7.073	*	* X

R denotes an observation with a large standardized residual
X denotes an observation whose X value gives it large influence.

Figure 3.6 Residuals versus Fits for M_0 bias for Model M_0

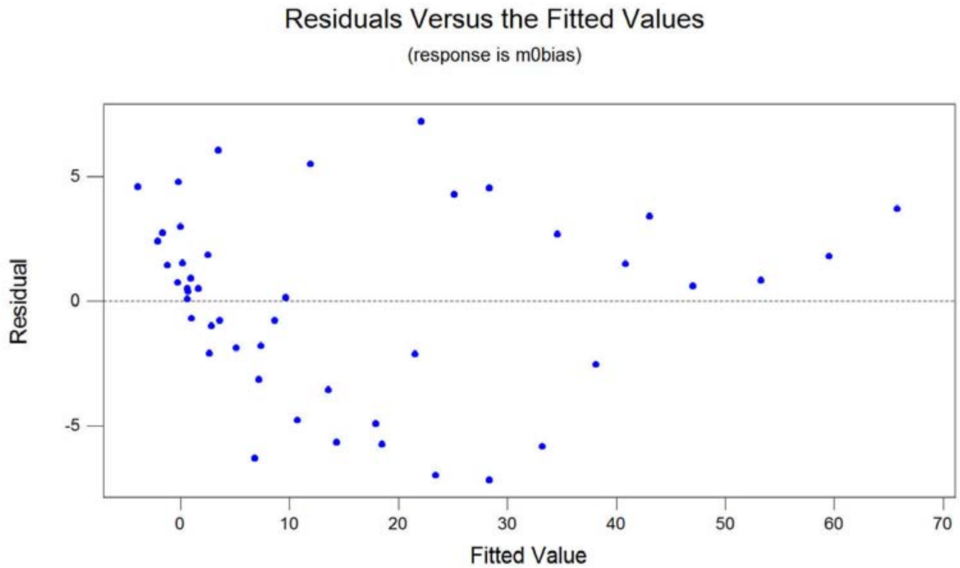


Figure 3.7 Normal Probability Plot of the Residuals for M_0 bias for Model M_0

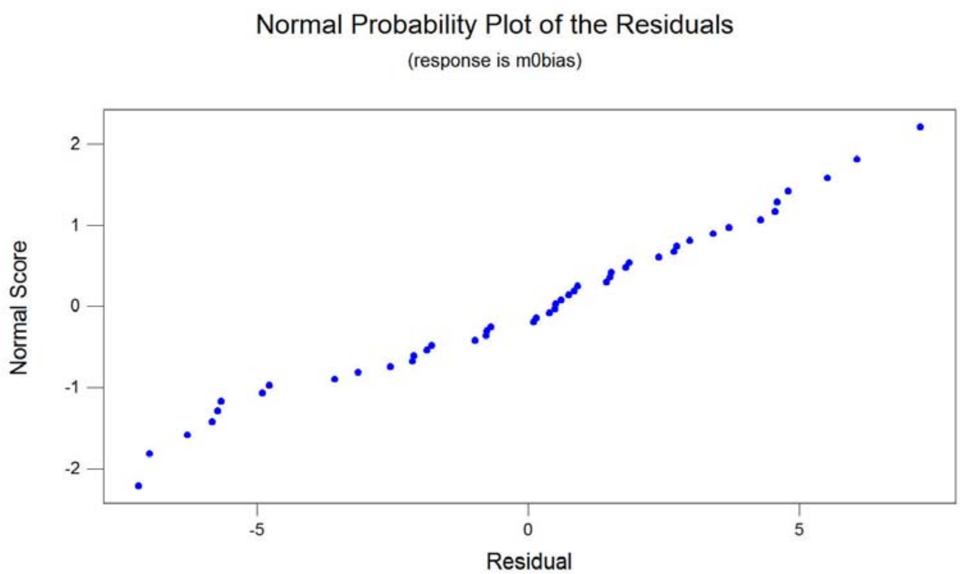


Figure 3.8 Regression Analysis Results for M_b bias in Model M_0

Regression Analysis: mbbias versus pcappct, pcapSq, nocc, nocc*pcap

```
The regression equation is
mbbias = 120 - 3.44 pcappct + 0.0242 pcapSq - 7.60 nocc + 0.127 nocc*pcap

46 cases used 18 cases contain missing values

Predictor        Coef     SE Coef          T        P
Constant      119.793       4.965      24.13    0.000
pcappct       -3.4422      0.2196     -15.67    0.000
pcapSq       0.024208    0.001964      12.33    0.000
nocc          -7.5967      0.6096     -12.46    0.000
nocc*pca      0.12697     0.01835       6.92    0.000

S = 3.891       R-Sq = 96.4%     R-Sq(adj) = 96.0%

Analysis of Variance

Source            DF          SS          MS         F        P
Regression         4     16464.6      4116.1    271.84    0.000
Residual Error    41       620.8        15.1
Total             45     17085.4

Source       DF      Seq SS
pcappct       1      9847.8
pcapSq        1      2913.6
nocc          1      2977.9
nocc*pca      1       725.2

Unusual Observations
Obs    pcappct     mbbias         Fit      SE Fit    Residual    St Resid
  8       80.0      0.534       7.025       2.596      -6.491     -2.24RX
 16       80.0          *       9.586       2.744           *         * X
 24       80.0          *      12.146       3.189           *         * X
 31       70.0          *       5.200       2.333           *         * X
 32       80.0          *      14.707       3.829           *         * X
 39       70.0          *       6.491       2.963           *         * X
 40       80.0          *      17.267       4.582           *         * X
 46       60.0          *       0.577       2.324           *         * X
 47       70.0          *       7.782       3.655           *         * X
 48       80.0          *      19.828       5.403           *         * X
 54       60.0          *       0.598       2.896           *         * X
 55       70.0          *       9.072       4.380           *         * X
 56       80.0          *      22.389       6.264           *         * X
 57       10.0     32.212      24.521       1.913       7.691       2.27R
 61       50.0          *      -4.284       2.247           *         * X
 62       60.0          *       0.619       3.484           *         * X
 63       70.0          *      10.363       5.123           *         * X
 64       80.0          *      24.949       7.150           *         * X

R denotes an observation with a large standardized residual
X denotes an observation whose X value gives it large influence.
```

Figure 3.9 Residuals versus Fits for M_b bias for Model M_0

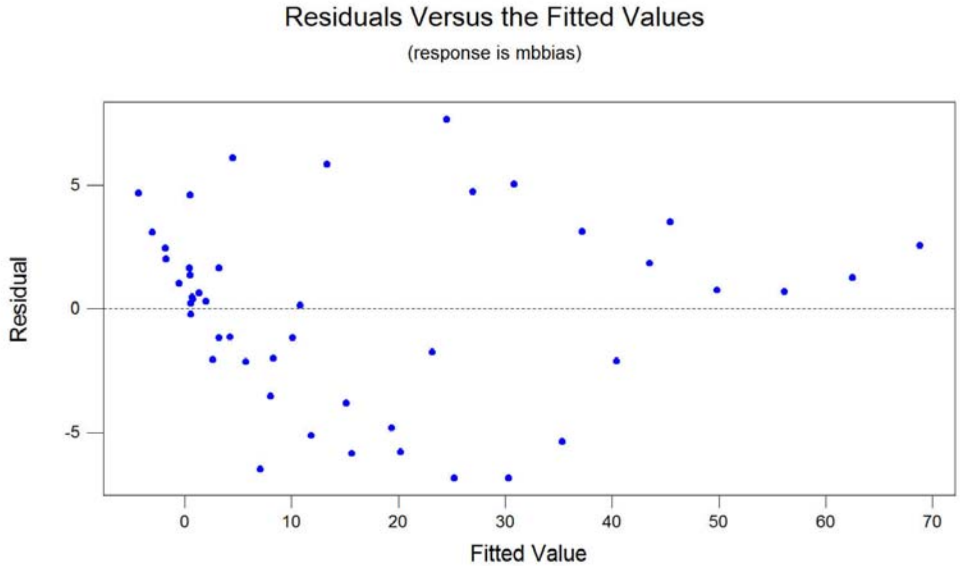


Figure 3.10 Normal Probability Plot of the Residuals for M_b bias for Model M_0

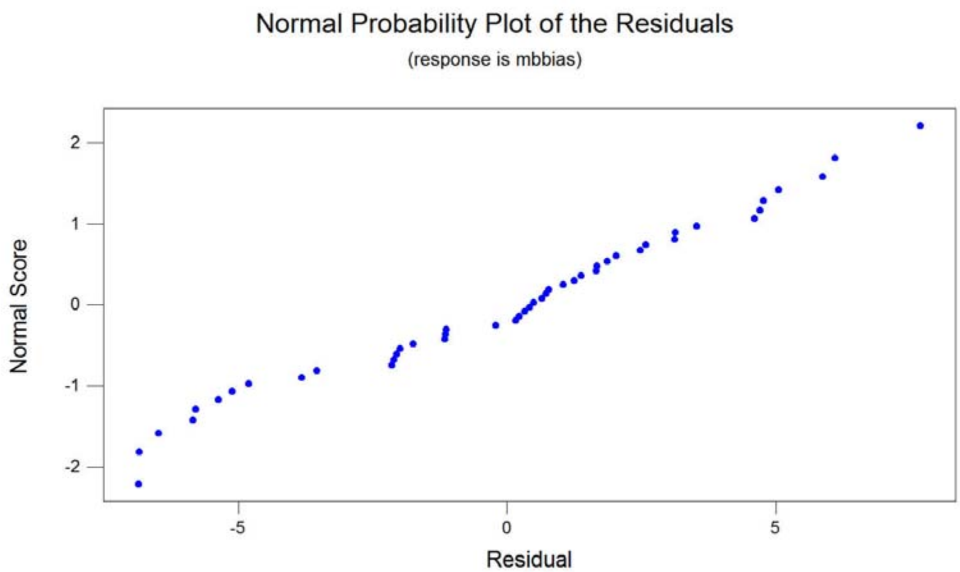


Table 3.3 Regression Results Examples for M_0 bias in Model M_0

nocc	pcappct	m0bias	m0bias fit
3	10	69.473	65.770
3	30	29.363	25.079
3	50	9.524	3.460
6	10	47.651	47.047
6	30	8.646	14.320
6	50	1.051	0.664
9	10	32.876	28.325
9	30	2.788	3.561
9	50	0.280	-2.132

Table 3.4 Regression Results Examples for M_b bias in Model M_0

nocc	pcappct	mbbias	mbbias fit
3	10	71.385	68.810
3	30	31.715	26.950
3	50	10.564	4.456
6	10	50.599	49.829
6	30	9.734	15.587
6	50	1.121	0.710
9	10	35.904	30.848
9	30	3.085	4.223
9	50	0.080	-3.035

From the regression analysis results, we first looked at the plots to determine whether the model that was selected was an appropriate model. Figures 3.7 and 3.10 are the normal plots of the residuals for the $\mathbf{M}_0$ and $\mathbf{M}_b$ bias respectively. Curvature in these plots would suggest that the model is not appropriate; however, these plots are acceptable since there is not much curvature. Figures 3.6 and 3.9 are plots of the residuals versus the fitted values. These plots are not random which casts doubt on the model. However, all terms included make logical sense given the behavior seen in Figures 3.1 and 3.2. From the regression models for $\mathbf{M}_0$ and $\mathbf{M}_b$ bias, it is noted that as probability of capture increases by 0.1, the bias decreases by about 3.4 percent, and as nocc increases by 1 the bias decreases by about 7.6 percent.

CHAPTER 4

BEHAVIOR-DEPENDENT CAPTURE PROBABILITIES

Section 4.1 Simulation of M_b Captures and Estimates Obtained

A program was written in Fortran to simulate a closed population with capture probabilities varying by behavioral response to capture. This program allowed us to specify the population size N and the number of repetitions of the simulation. The initial capture probability was varied from 0.1 to 0.8, and the number of capture occasions was varied from 3 to 10. In addition to these conditions, the probability of recapture was constrained between 0.1 and 0.8 and was varied from 0.3 below the probability of capture to 0.3 above the probability of capture. The probability of recapture did not equal the probability of capture in any of the cases, since this would be an $\mathbf{M}_0$ capture model which was discussed in Chapter 3. From the simulated capture data, maximum likelihood estimates of the population size were computed for estimators $\mathbf{M}_0$, $\mathbf{M}_b$ and $\mathbf{M}_t$. If all three are equivalent they should result in similar estimates of the population. To analyze the simulated results, graphics were generated to look for patterns.

Figure 4.1 Model M_b Capture Occasion Effect for Capture Probability p = 0.1

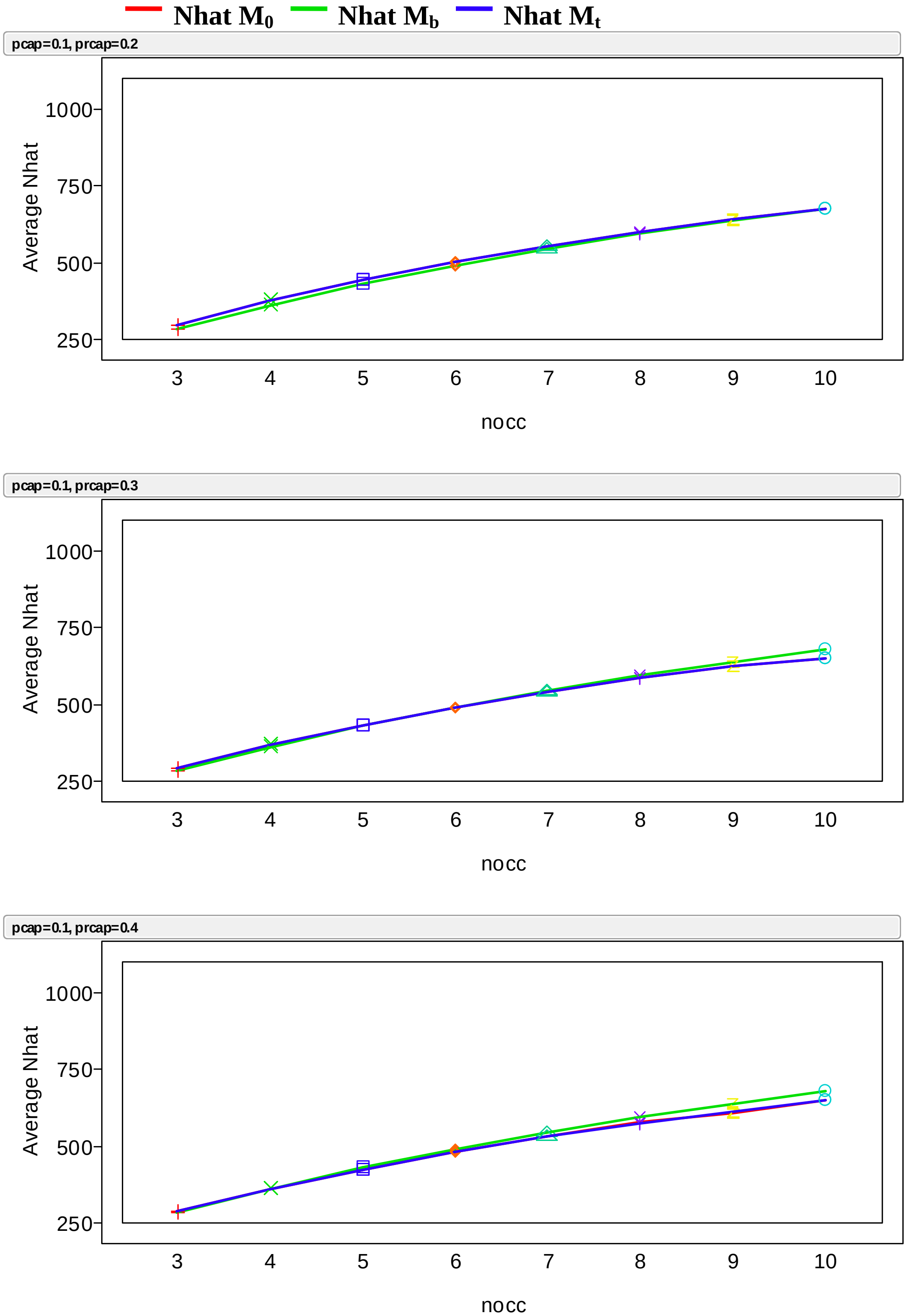


Figure 4.2 Model M_b Capture Occasion Effect for Capture Probability p = 0.2

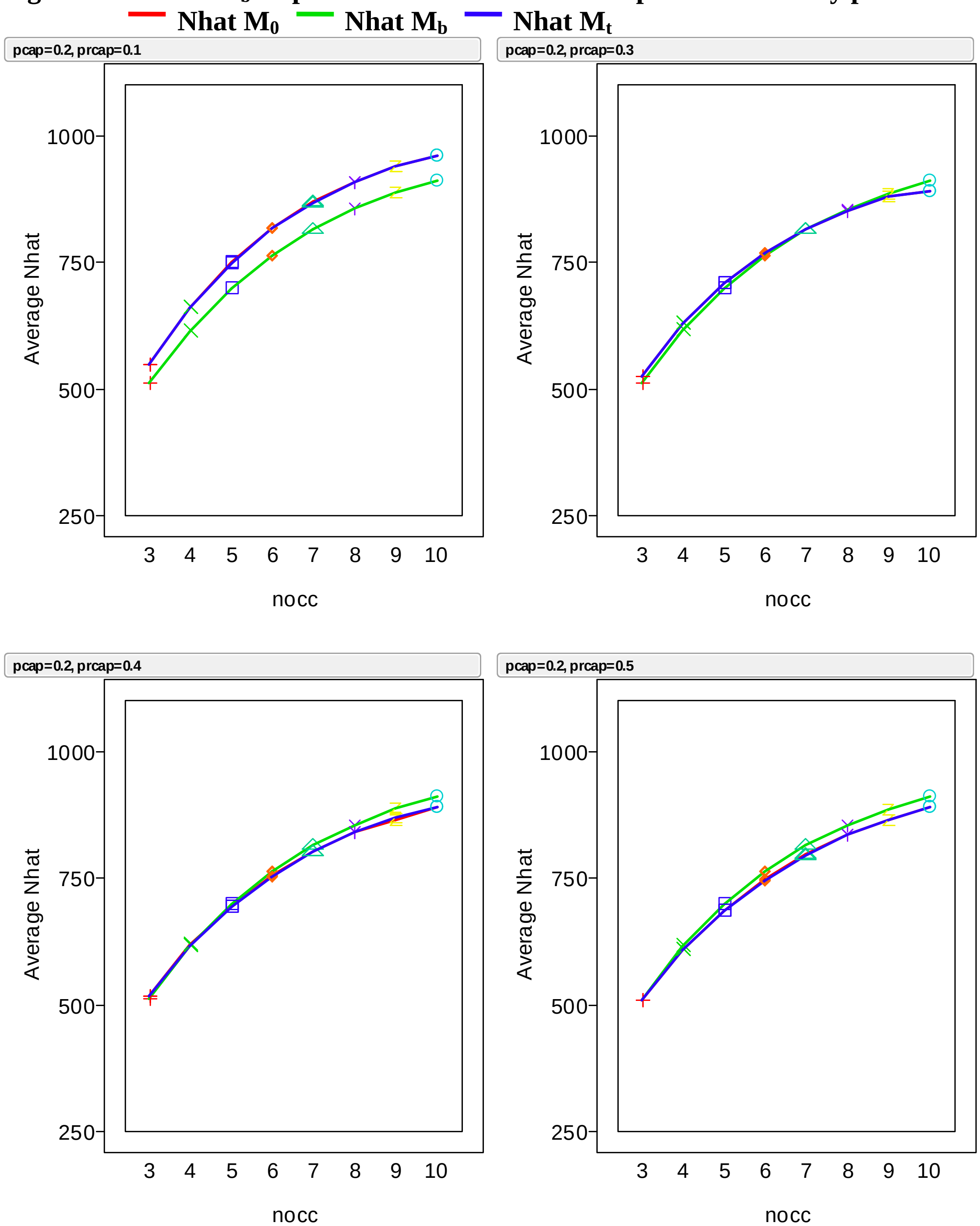


Figure 4.3 Model M_b Capture Occasion Effect for Capture Probability p = 0.3

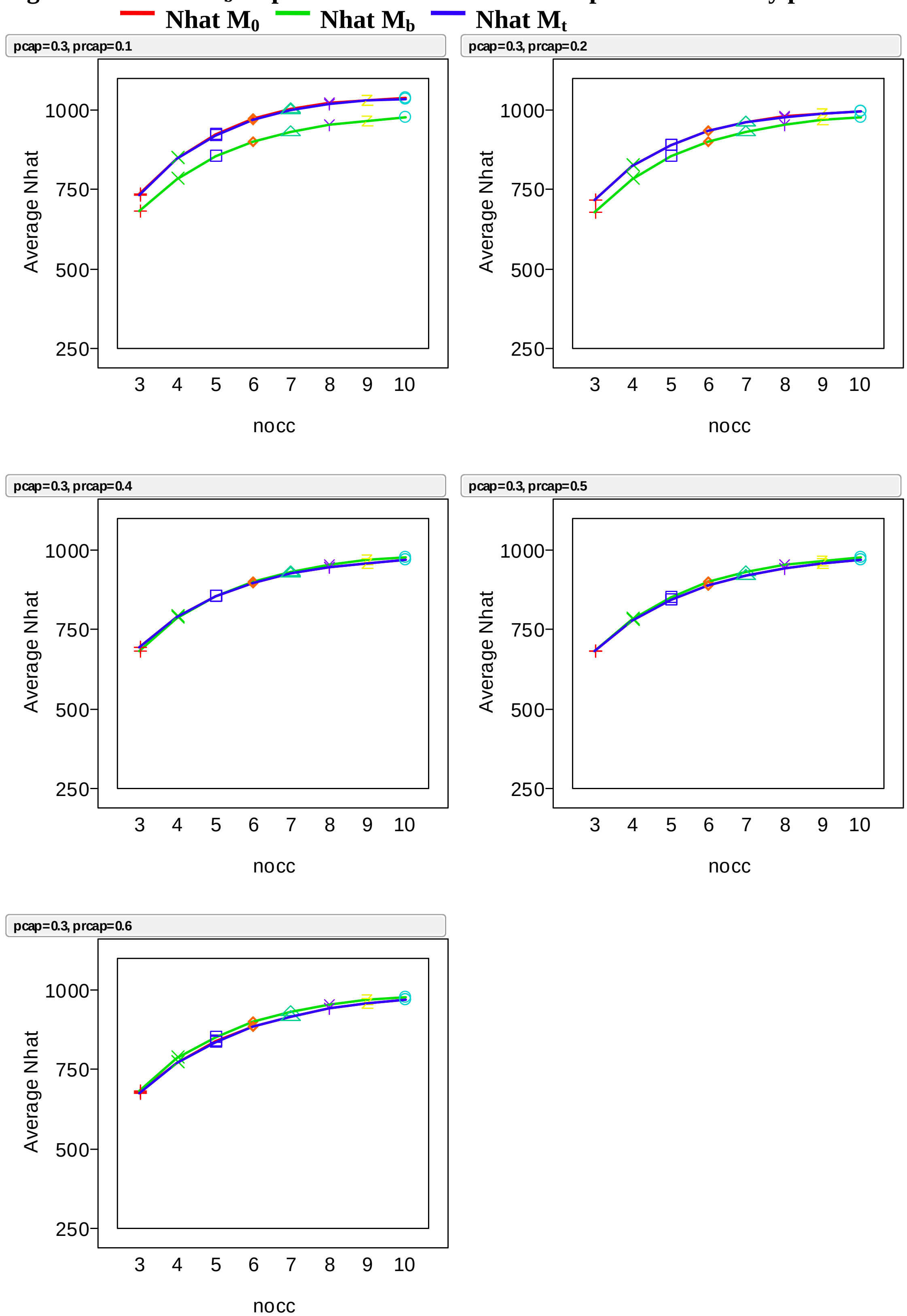


Figure 4.4 Model M_b Capture Occasion Effect for Capture Probability p = 0.4

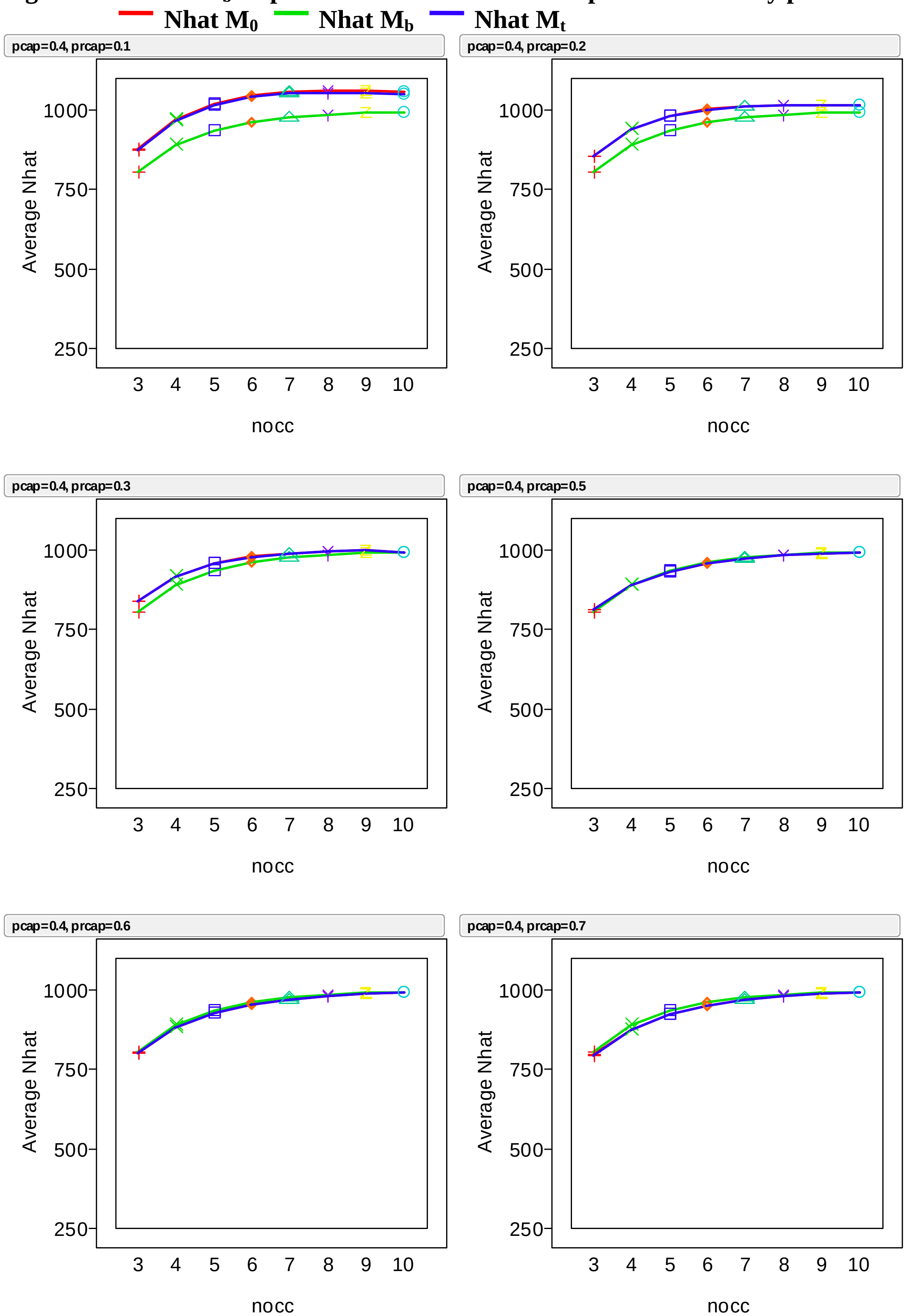


Figure 4.5 Model M_b Capture Occasion Effect for Capture Probability p = 0.5

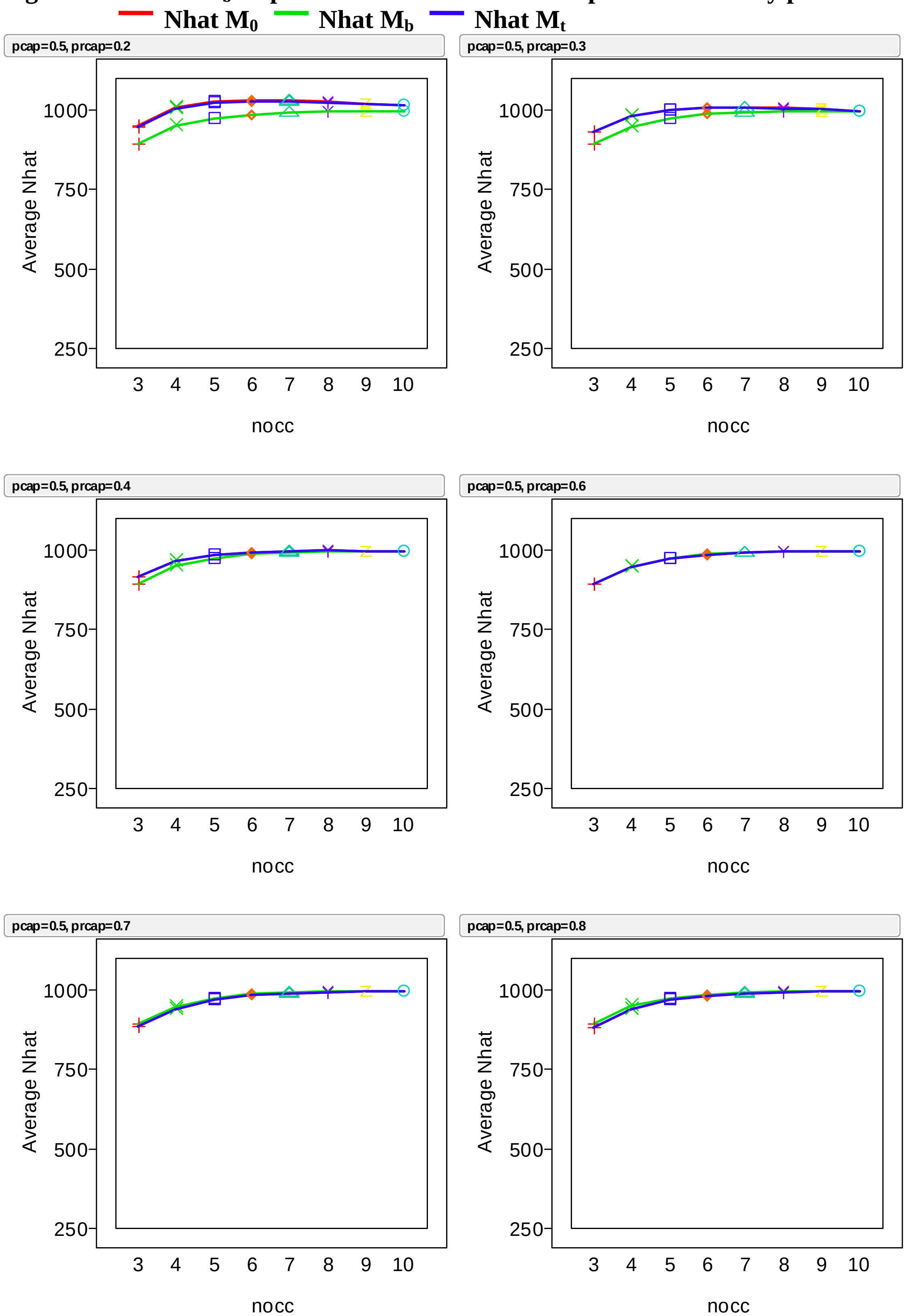


Figure 4.6 Model M_b Capture Occasion Effect for Capture Probability p = 0.6

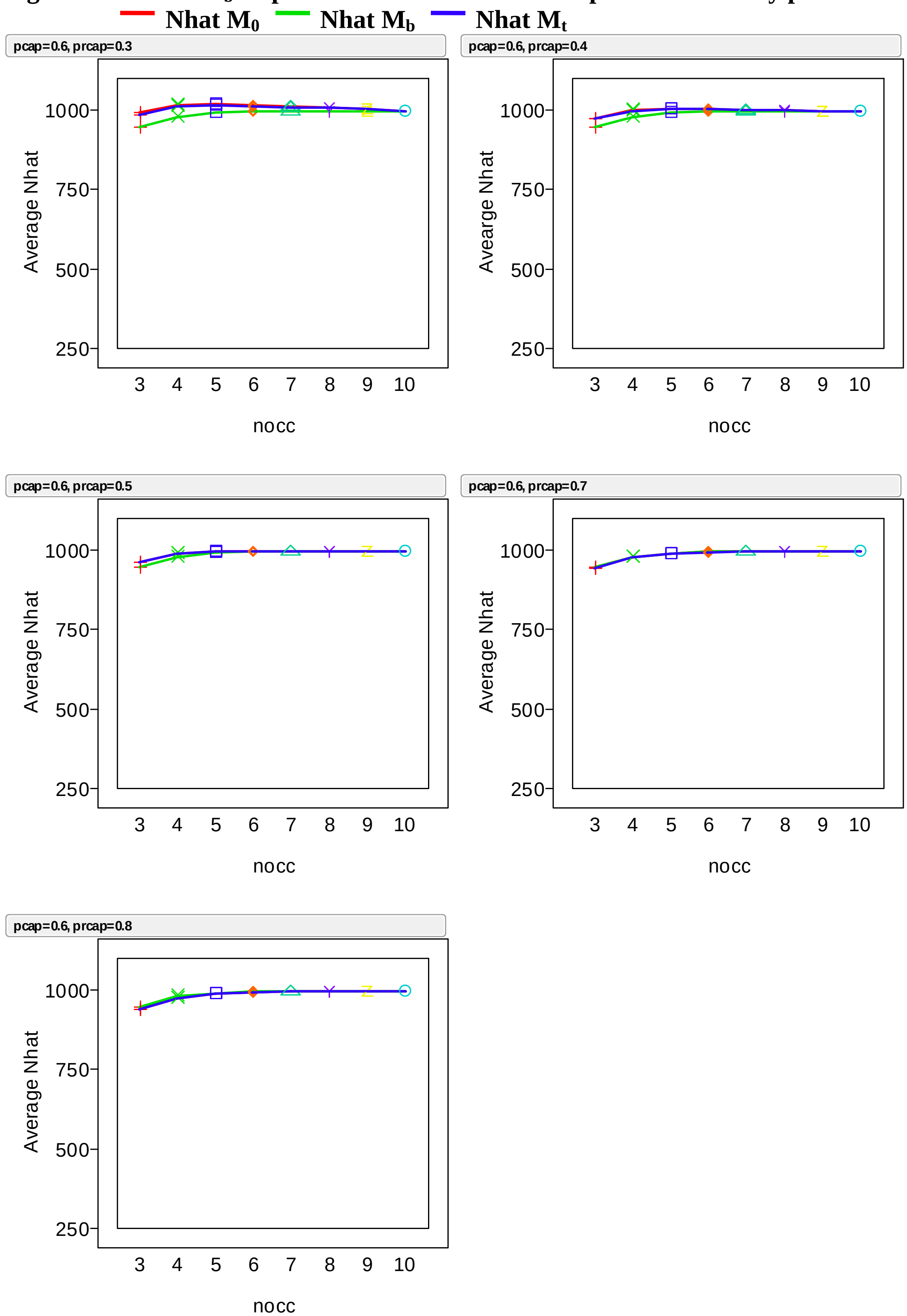


Figure 4.7 Model M_b Capture Occasion Effect for Capture Probability p = 0.7

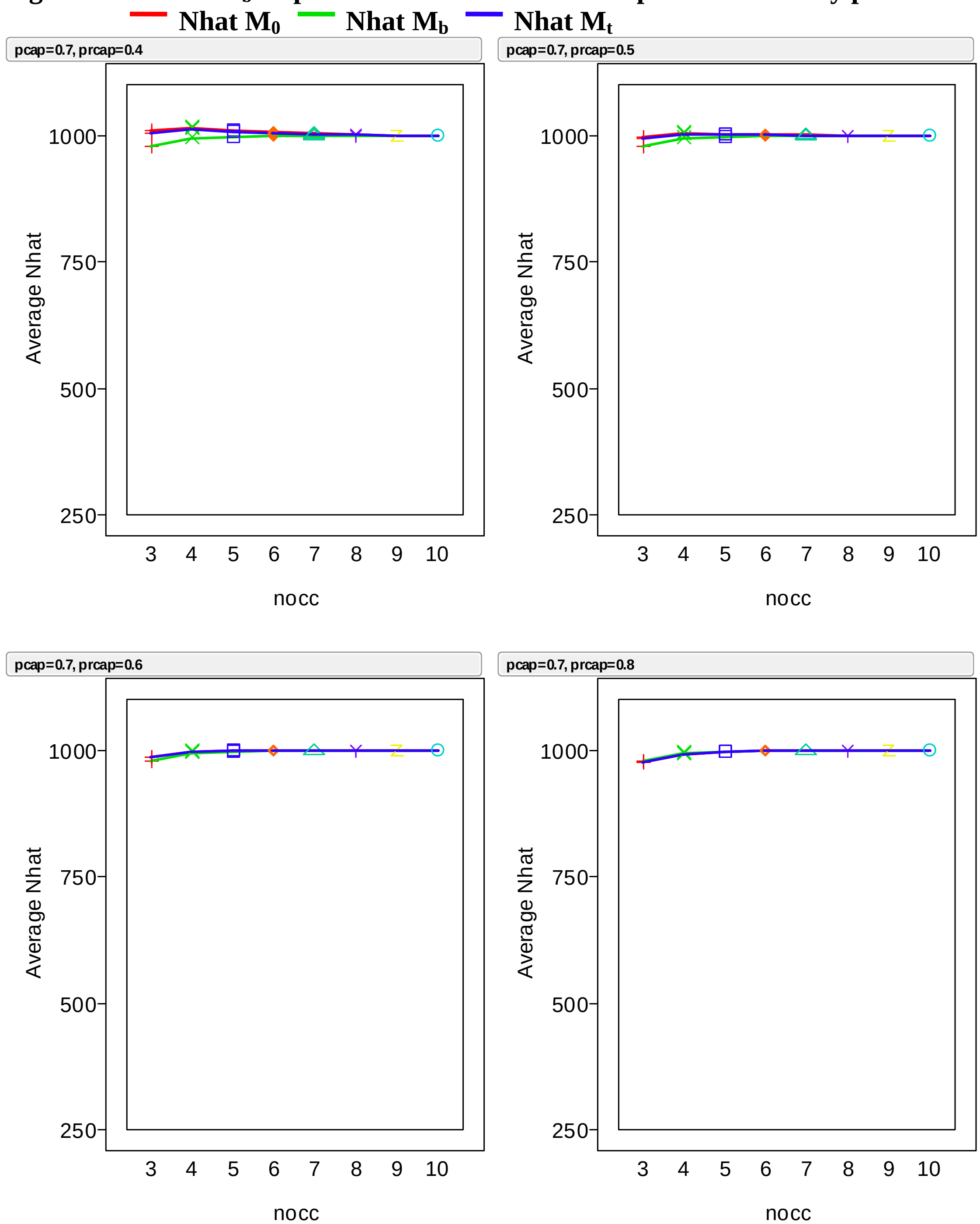
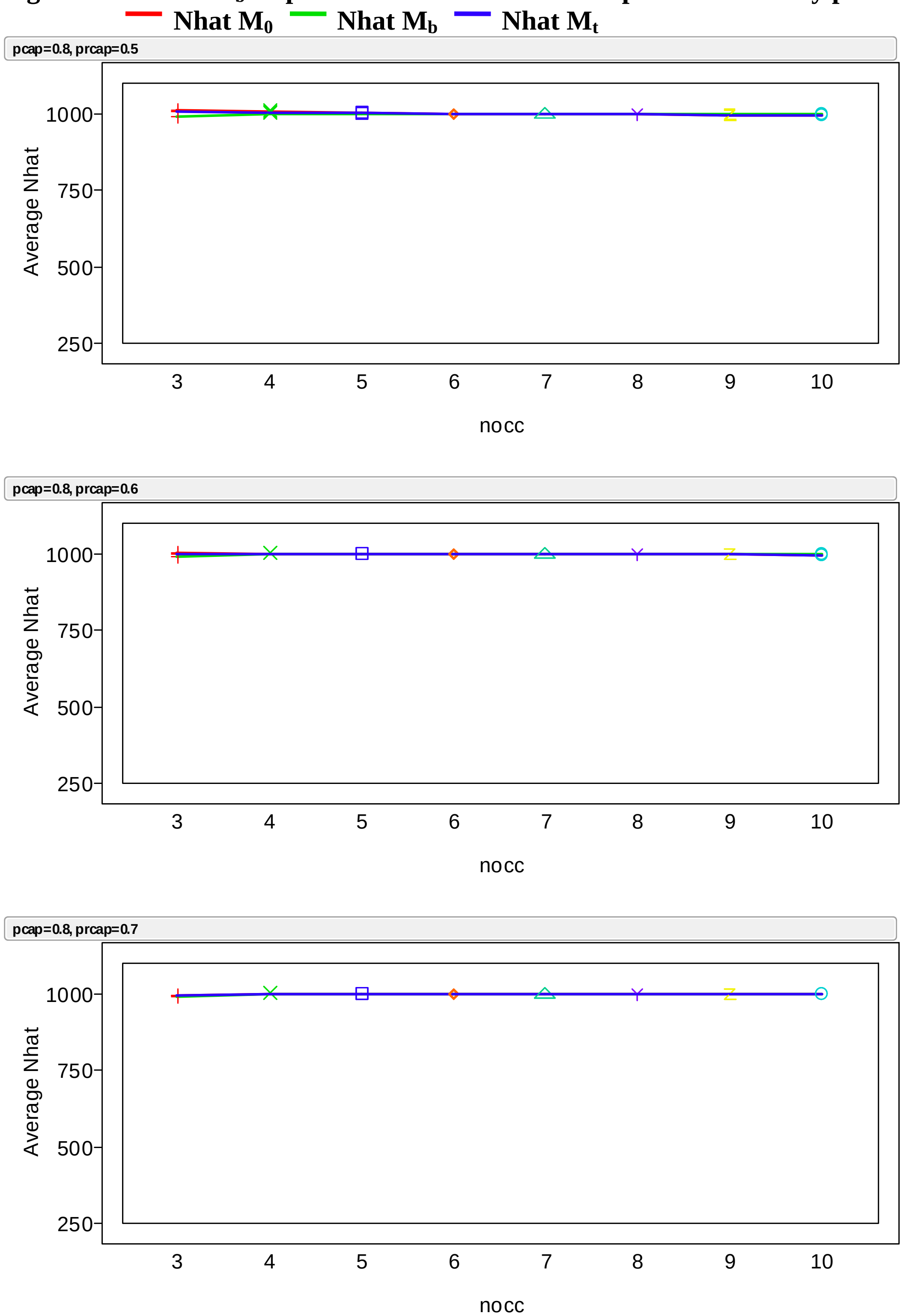


Figure 4.8 Model M_b Capture Occasion Effect for Capture Probability p = 0.8



Each of the figures shows the estimates of the population for combinations of capture probability and recapture probability. In most of the figures it is seen that when the probability of recapture is less than the probability of capture, the $\mathbf{M}_b$ estimates are lower than those of $\mathbf{M}_0$ and $\mathbf{M}_t$, which are essentially the same. As the probability of recapture becomes higher than the probability of capture, the estimates in $\mathbf{M}_b$ become higher than those of $\mathbf{M}_0$ and $\mathbf{M}_t$. As the probability of capture becomes higher, the estimates for all of the models are essentially the same, and they are very close to the actual population size. For the low capture and recapture probabilities, the estimates for all of the models are negatively biased. In all cases, as the number of capture occasions (nocc) increases, the estimates become closer to the actual population.

It is interesting to note that average population estimates were above the actual population levels about 7 percent of the time for $\mathbf{M}_0$ and $\mathbf{M}_t$ population estimators. This occurred when the initial capture probability was at least 0.2 larger than the recapture probability (a "trap shy" situation). The converse of this is also seen. The lowest population estimates as a percent of the actual population for the $\mathbf{M}_0$ and $\mathbf{M}_t$ estimators are seen when recapture probabilities are much higher than initial capture probabilities (a "trap happy" situation). This behavior can be partially explained in how the different estimators treat recaptures. The $\mathbf{M}_0$ and $\mathbf{M}_t$ estimators focus primarily on the number of captured animals, whereas the $\mathbf{M}_b$ estimator focuses on the number of marked animals assumed to be in the population at the time. Recaptures do not convey information about the population size for the $\mathbf{M}_b$ estimator.

Section 4.2 Bias Modeling

Figures 4.9 and 4.12 are the results from a regression analysis which was performed in Minitab to model the bias for the $\mathbf{M}_0$ and $\mathbf{M}_b$ estimators. As in chapter 3, the $\mathbf{M}_0$ and $\mathbf{M}_t$ models were essentially the same, so we did not model them separately. The data set was trimmed prior to performing this analysis. We removed simulation cases after the MLE had essentially no bias (the MLE estimated at least 99 percent of the actual population) for a given number of captures, to reduce the weight these cases had on the regression. The values that were trimmed are listed in Table 4.1.

Table 4.1 Criteria for Data Trimming of Model $\mathbf{M}_b$ Results

nocc	pcap greater than or equal to
4	0.8
5	0.7
6	0.7
7	0.6
8	0.6
9	0.6
10	0.5

Figure 4.9 Regression Analysis Results for M_0 bias in Model M_b

Regression Analysis: m0bias versus pcappct, pcapSq, nocc, nocc*pcap, pdiffpct

```
The regression equation is
m0bias = 116 - 3.45 pcappct + 0.0253 pcapSq - 7.44 nocc + 0.128 nocc*pcap
           - 0.121 pdiffpct

217 cases used 71 cases contain missing values

Predictor        Coef     SE Coef          T        P
Constant      115.824       2.588      44.76    0.000
pcappct       -3.4522      0.1066     -32.38    0.000
pcapSq      0.0252706   0.0009536      26.50    0.000
nocc          -7.4427      0.3109     -23.94    0.000
nocc*pca     0.127561    0.008682      14.69    0.000
pdiffpct     -0.12084     0.01403      -8.62    0.000

S = 3.813       R-Sq = 95.3%     R-Sq(adj) = 95.2%

Analysis of Variance

Source            DF          SS          MS         F        P
Regression         5       62852       12570    864.72    0.000
Residual Error   211        3067          15
Total            216       65920

Source       DF      Seq SS
pcappct       1       36039
pcapSq        1       12385
nocc          1       10134
nocc*pca      1        3216
pdiffpct      1        1079
```

Figure 4.10 Residuals versus Fits for M_0 bias for Model M_b

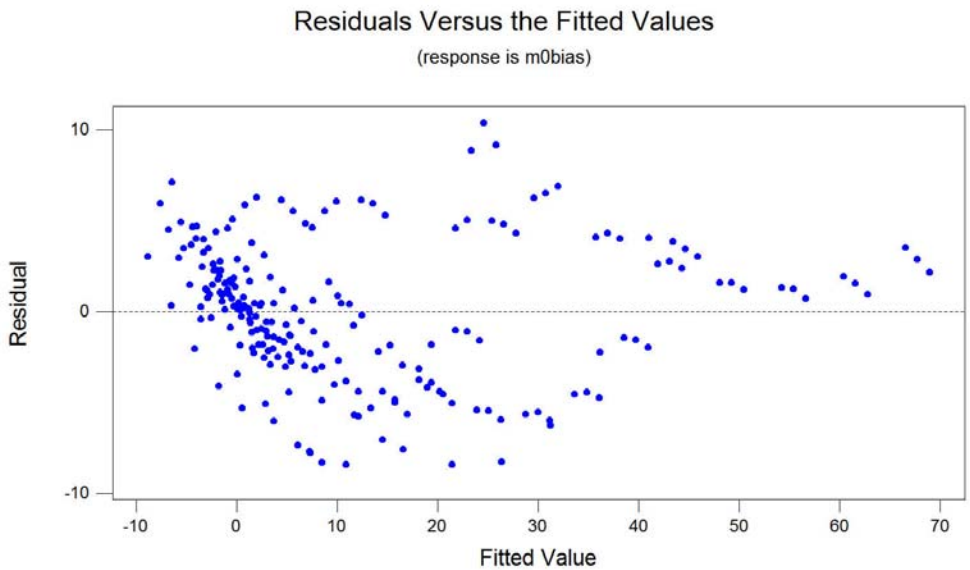


Figure 4.11 Normal Probability Plot of the Residuals for M_0 bias for Model M_b

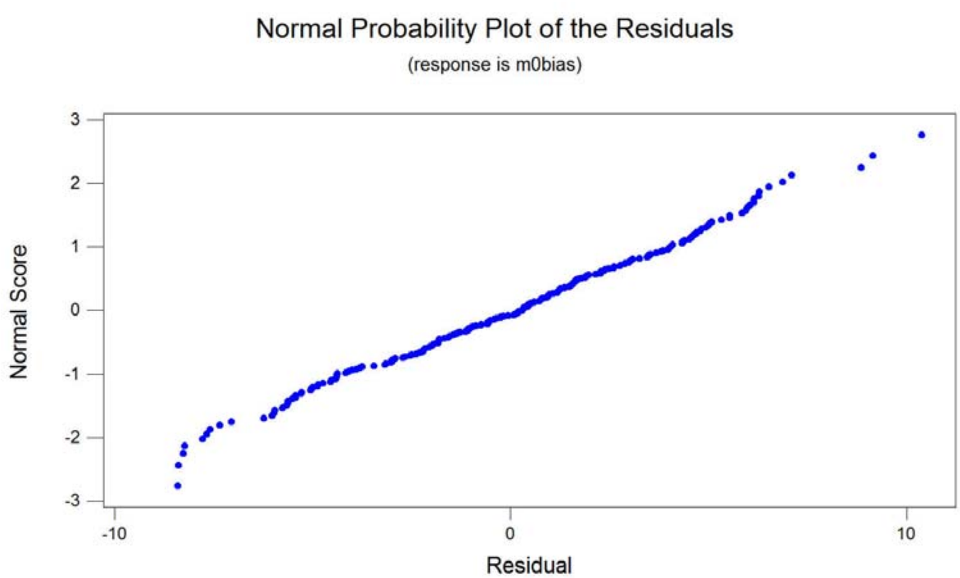


Figure 4.12 Regression Analysis Results for M_b bias in Model M_b

Regression Analysis: mbbias versus pcappct, pcapSq, nocc, nocc*pcap

```
The regression equation is
mbbias = 119 - 3.42 pcappct + 0.0240 pcapSq - 7.73 nocc + 0.130 nocc*pcap

217 cases used 71 cases contain missing values

Predictor        Coef     SE Coef          T        P
Constant      119.448       2.385      50.09    0.000
pcappct      -3.41602     0.09887     -34.55    0.000
pcapSq      0.0240015   0.0008884      27.02    0.000
nocc          -7.7278      0.2899     -26.66    0.000
nocc*pca     0.129714    0.008095      16.02    0.000

S = 3.556       R-Sq = 95.7%     R-Sq(adj) = 95.6%

Analysis of Variance

Source            DF          SS          MS         F        P
Regression         4       60087       15022   1188.12    0.000
Residual Error   212        2680          13
Total            216       62767

Source       DF      Seq SS
pcappct       1       34387
pcapSq        1       10931
nocc          1       11522
nocc*pca      1        3246
```

Figure 4.13 Residuals versus Fits for M_b bias for Model M_b

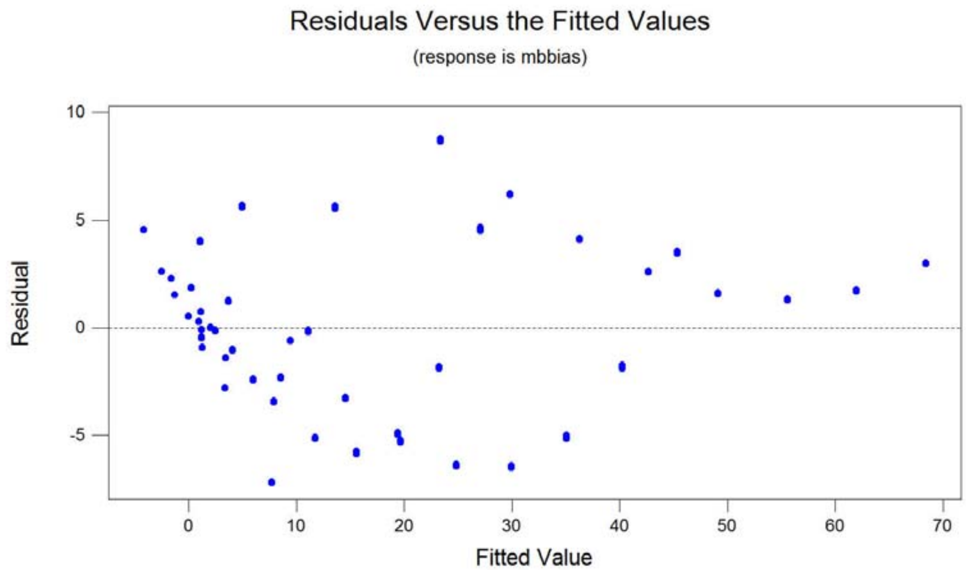


Figure 4.14 Normal Probability Plot of the Residuals for M_b bias for Model M_b

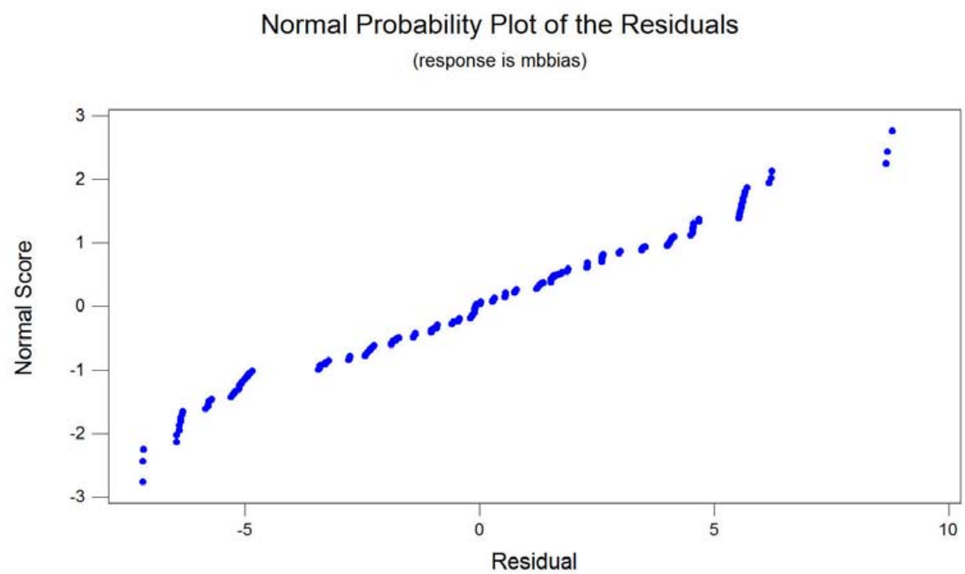


Table 4.2 Regression Results Examples for M_0 bias in Model M_b

nocc	pcappct	rcappct	pdiffpct	m0bias	m0bias fit
3	10	20	-10	70.075	66.537
3	10	30	-20	70.612	67.745
3	10	40	-30	71.094	68.954
5	10	20	-10	55.544	54.203
5	10	30	-20	56.642	55.411
5	10	40	-30	57.347	56.619
7	10	20	-10	44.468	41.868
7	10	30	-20	45.838	43.077
7	10	40	-30	46.667	44.285
9	10	20	-10	35.798	29.534
9	10	30	-20	37.267	30.743
9	10	40	-30	38.835	31.951
3	30	20	10	27.986	22.946
3	30	40	-10	30.368	25.363
3	30	60	-30	32.110	27.780
5	30	20	10	10.705	15.715
5	30	40	-10	14.371	18.132
5	30	60	-30	16.017	20.548
7	30	20	10	3.602	8.483
7	30	40	-10	7.075	10.900
7	30	60	-30	8.031	13.317
9	30	20	10	0.818	1.252
9	30	40	-10	4.104	3.668
9	30	60	-30	4.109	6.085
3	50	20	30	4.658	-0.427
3	50	30	20	6.639	0.781
3	50	70	-20	11.154	5.615
3	50	80	-30	11.653	6.823
5	50	20	30	-2.910	-2.557
5	50	30	20	-0.373	-1.348
5	50	70	-20	2.933	3.486
5	50	80	-30	3.015	4.694
7	50	20	30	-3.194	-4.686
7	50	30	20	-0.999	-3.477
7	50	70	-20	0.763	1.356
7	50	80	-30	0.763	2.565
3	70	40	30	-1.065	-1.168
3	70	50	20	0.235	0.041

Table 4.3 Regression Results Examples for M_b bias in Model M_b

nocc	pcappct	mbbias	mbbias fit
3	10	71.368	68.396
3	30	31.734	27.059
3	50	10.541	4.924
6	10	50.740	49.104
6	30	9.843	15.550
6	50	1.102	1.198
9	10	36.024	29.812
9	30	3.049	4.041
9	50	0.084	-2.529

As we did in Chapter 3, we first looked at the plots that were obtained from the regression analysis to determine if the model was appropriately selected. The normal plots of the residuals, Figures 4.11 and 4.14, have a little curvature on the ends, but considering the sample size, the model selected should be appropriate. Figures 4.10 and 4.13 are the residuals versus the fitted values. Again, as we saw in Chapter 3, these plots are not perfectly random, but all the terms that are included in the model make sense based on the observations made from the plots obtained from the simulated data. From the regression models for $\mathbf{M}_0$ and $\mathbf{M}_b$ bias, it is noted that as probability of capture increases by 0.1, the bias decreases by about 3.4 percent. As nocc increases by 1, the bias decreases by about 7.4 percent for $\mathbf{M}_0$ and it decreases by about 7.7 percent for $\mathbf{M}_b$.

CHAPTER 5

TIME-DEPENDENT CAPTURE PROBABILITIES

Three programs were written in Fortran to simulate a closed population with capture probabilities varying with time. Each of the three programs was written for a specific number of capture occasions: three, four and five. The programs allowed us to specify the population size N and the number of repetitions of the simulation. In each program the probability of capture on the first occasion was varied from 0.1 to 0.8. The other capture probabilities varied from 0.5 below that to 0.5 above, with all capture probabilities not the same, as this would be the $\mathbf{M}_0$ capture scenario. From the simulated capture data, maximum likelihood estimates of the population size were computed for estimators $\mathbf{M}_0$, $\mathbf{M}_b$ and $\mathbf{M}_t$. To compare estimators for the different models, graphics were produced similar to those seen in Chapters 3 and 4. However, due to the sheer number of possible capture probability combinations, a variety of interesting combinations were selected to further examine to see what, if any, patterns were present with capture probability changes, and how these were similar to or different from those already examined.

As stated previously, this capture scenario comes into play in several practical instances, including wildlife situations where capture effort is not constant across capture occasions, or when weather affects the availability of individuals in the population to be captured; in the case of human populations, this occurs when membership in one group (list) is less likely than in others.

Section 5.1 Simulation of M_t Captures for Three Capture Occasions

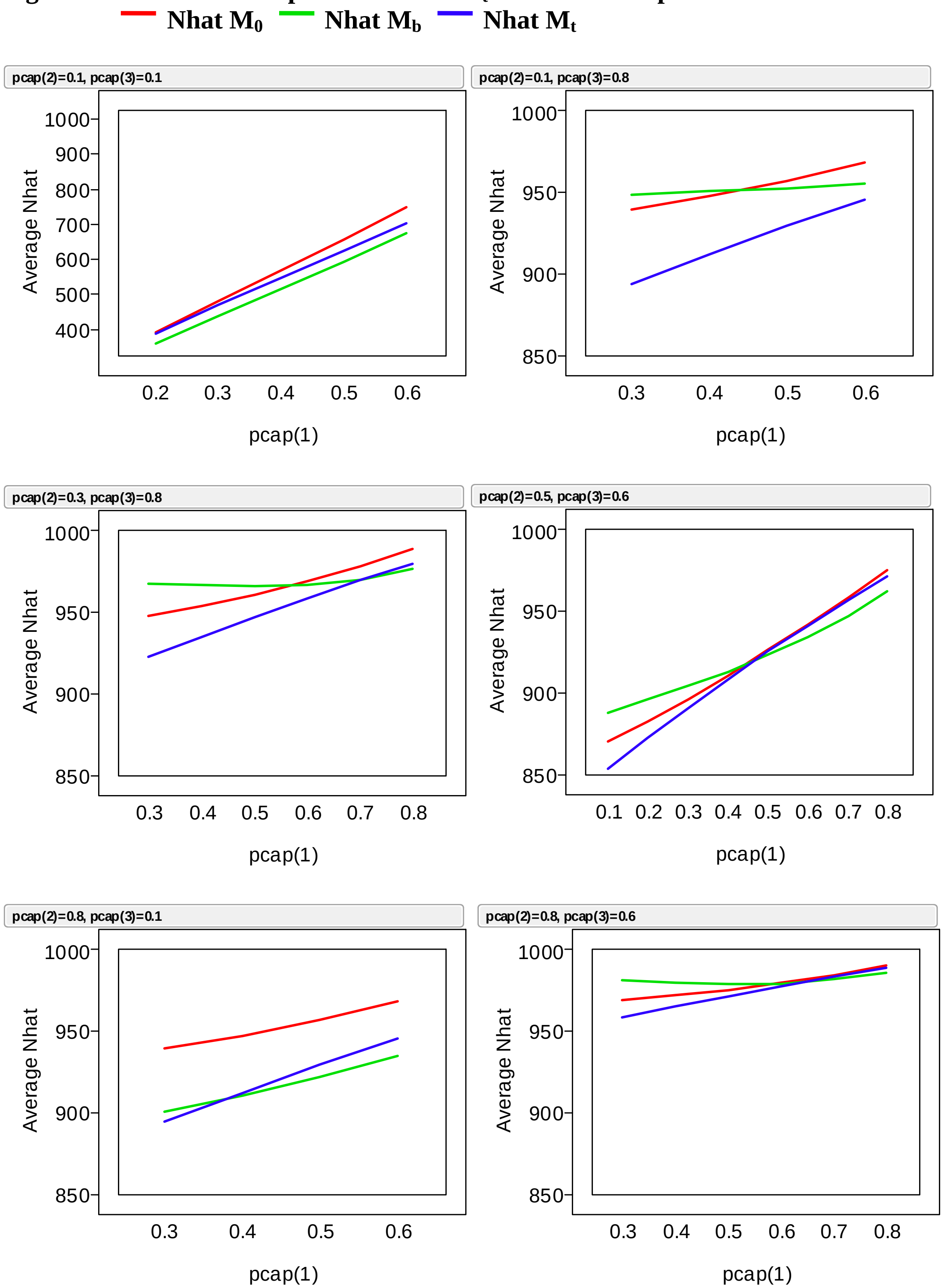
The first case that we looked at was for three capture occasions. Graphics were generated from the simulated data to compare the estimated values for $\mathbf{M}_0$, $\mathbf{M}_b$ and $\mathbf{M}_t$. Figure 5.1 shows how the different estimators behave for changing probability of capture on the first capture occasion for various combinations of probability of capture on the second and third capture occasions.

In the first panel, we see that for three capture occasions, when the probability of capture is equal to 0.1 on both the second and third capture occasions, all of the estimators give low estimates of the population, with the $\mathbf{M}_b$ estimator giving the lowest estimates of the three. Further note that the behavior appears linear, unlike the marked negative concavity seen previously.

For probability of capture equal to 0.1 on the second capture occasion and probability of capture equal to 0.8 on the third capture occasion, we see that the $\mathbf{M}_t$ estimator is now giving the lowest, most biased estimate, and the $\mathbf{M}_b$ estimator is higher than the other two when the probability of capture on the first capture occasion is low. As the probability of capture for the first capture occasion increases, the $\mathbf{M}_b$ estimator crosses the $\mathbf{M}_0$ estimator. The values for the $\mathbf{M}_b$ estimator then lie between those of the $\mathbf{M}_t$ and $\mathbf{M}_0$ estimators.

The graphic for the probability of capture equal to 0.3 on the second capture occasion and probability of capture equal to 0.8 on the third capture occasion also shows the $\mathbf{M}_b$ estimator being the highest and the $\mathbf{M}_t$ estimator being the lowest when the probability of capture on the first capture occasion is low. We also begin to see some curvature for the $\mathbf{M}_b$ estimator; however, this curvature is opposite that seen with the

Figure 5.1 Selected Examples for Model M_t with Three Capture Occasions



other capture scenarios. The same positive curvature is also seen for the $\mathbf{M}_b$ estimator when the probability of capture is equal to 0.5 on the second capture occasion and the probability of capture is equal to 0.6 on the third capture occasion. For this particular combination, the $\mathbf{M}_b$ estimator gives the highest values when the capture probability on the first capture occasion is the lowest. As the capture probability increases on the first capture occasion, the $\mathbf{M}_0$ and $\mathbf{M}_t$ estimators merge together, and the $\mathbf{M}_b$ estimator crosses these two and gives the lowest estimates at higher first occasion capture probabilities.

The next combination that we chose to look at for three capture occasions was one where the probability of capture on the second capture occasion was high and the probability of capture on the third capture occasion was low. The graphic clearly shows that for this case, the $\mathbf{M}_0$ estimator gives the least biased estimate as the probability of capture on the first capture occasion increases. The $\mathbf{M}_b$ and $\mathbf{M}_t$ estimators give similar estimates, with the $\mathbf{M}_t$ estimator giving a slightly better estimate as the probability of capture on the first capture occasion increases. A possible cause for the $\mathbf{M}_b$ estimator giving the lowest estimates for the higher capture probabilities on the first occasion is there are probably not many unmarks left to give it additional information on N.

The last graph in Figure 5.1 shows how the three estimators behave when the probability of capture on the second capture occasion is 0.8 and the probability of capture on the third capture occasion is 0.6. We see that for a lower probability of capture on the first capture occasion, the three estimators are slightly separated in their estimates, with $\mathbf{M}_b$ giving the highest estimate and $\mathbf{M}_t$ giving the lowest estimate. As the capture probability on the first capture occasion becomes higher, the three estimators begin to merge together; however it can be seen that the $\mathbf{M}_b$ estimator is just slightly lower than the other two. We also noted that out of the graphics we selected, this combination has the initial estimates closest to the actual population of 1000 for all three of the estimators.

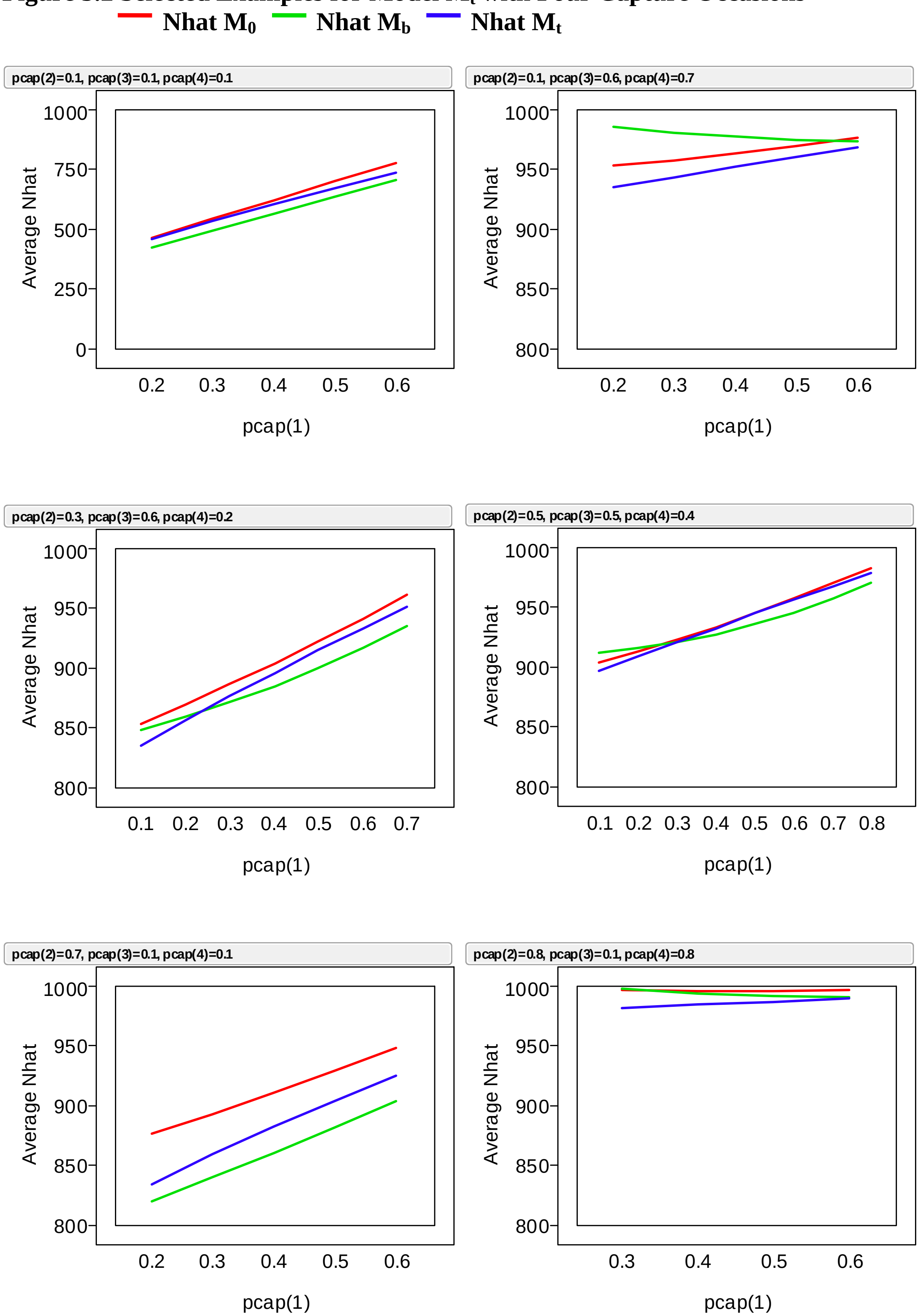
Section 5.2 Simulation of $\mathbf{M_t}$ Captures for Four Capture Occasions

We then looked at how the estimators behaved when there were four capture occasions. As we did for three capture occasions, we generated graphics to see how the three estimators behave on the first capture occasion for various combinations of the capture probability on the second, third and fourth capture occasion. We only chose a select few combinations which are in Figure 5.2.

In the first panel, we see that for four capture occasions, when the probability of capture is equal to 0.1 on the second, third and fourth capture occasions, all of the estimators again give low estimates of the population, with the $\mathbf{M_b}$ estimator giving the lowest estimates of the three. As we saw for three capture occasions, the behavior appears linear, unlike the marked negative concavity seen for the models discussed in Chapters 3 and 4. Also, $\mathbf{M_0}$ and $\mathbf{M_t}$ start out giving very close estimates, but as the capture probability increases for the first capture occasion, they begin to separate.

We also looked at a combination of capture probabilities with the probability of capture being low on the second capture occasion and high on the third and fourth capture occasions. In this case where the probability of capture on the second occasion is 0.1, the probability of capture on the third occasion is 0.6 and the probability of capture on the fourth occasion is 0.7, we see that $\mathbf{M_b}$ has the highest initial estimate, and continues to have the highest estimate as the capture probability increases on the first capture occasion. $\mathbf{M_t}$ gives the lowest estimates, and $\mathbf{M_0}$ has values between $\mathbf{M_t}$ and $\mathbf{M_b}$.

Figure 5.2 Selected Examples for Model M_t with Four Capture Occasions



As the probability of capture on the first occasion reaches its highest value, $\mathbf{M}_b$ decreases and merges with $\mathbf{M}_0$.

Another combination we looked at was where the probability of capture on the second and fourth occasions was low and the probability of capture on the third occasion was high. For this case, the probability of capture was 0.3 on the second occasion, 0.6 on the third occasion and 0.2 on the fourth occasion. $\mathbf{M}_0$ gives the highest estimates of the population in this case. When the probability of capture on the first occasion is low, $\mathbf{M}_b$ gives an estimate between the estimates of $\mathbf{M}_t$ and $\mathbf{M}_0$. However, as the probability of capture on the first occasion increases, $\mathbf{M}_b$ curves upward and gives the lowest estimates.

For the capture probability equal to 0.5 on the second and third capture occasions, and 0.4 on the fourth capture occasion, we see a similar pattern of the $\mathbf{M}_b$ estimate being concave up as the capture probability on the first occasion increases. In this particular case the $\mathbf{M}_b$ estimator starts out giving the highest estimate, but as the probability of capture on the first capture occasion increases, $\mathbf{M}_b$ gives the lowest estimates. We also note that $\mathbf{M}_0$ and $\mathbf{M}_t$ give very similar estimates for all capture probabilities on the first capture occasion.

For a high capture probability on the second capture occasion and low capture probabilities on the third and fourth capture occasions, we see a linear pattern similar to the one where the capture probabilities were low for the second, third and fourth capture occasions. This is seen in the graphic where the probability of capture on the second occasion is equal to 0.7 and the probability of capture on the third and fourth capture occasions is equal to 0.1. We see that the three estimators all give different values as the probability of capture on the first occasion increases, with none of the estimates crossing or merging. $\mathbf{M}_b$ gives the lowest estimates and $\mathbf{M}_0$ gives the highest estimates as the probability of capture on the first capture occasion increases. $\mathbf{M}_t$ gives estimates that lie between $\mathbf{M}_b$ and $\mathbf{M}_0$.

The last combination that we chose to look at was for the probability of capture being low on the third capture occasion and high on the second and fourth capture occasions. The graphic for this case has the probability of capture on the second and fourth capture occasions equal to 0.8 and the probability of capture on the third occasion equal to 0.1. We see that the $\mathbf{M}_b$ estimator has positive concavity, but decreases with increasing first capture probability. $\mathbf{M}_b$ has the highest estimate at the lowest probability of capture on the first occasion, and then curves downward to merge with the $\mathbf{M}_t$ estimator to give the lowest estimate at the highest capture probability on the first capture occasion. $\mathbf{M}_0$ gives the highest estimate at the highest capture probability on the first capture occasion. We also note that when comparing this graphic to the other graphics that we chose to look at for four capture occasions, the three estimators for this combination of probabilities are the closest to the actual population of 1000.

Section 5.3 Simulation of $\mathbf{M}_t$ Captures for Five Capture Occasions

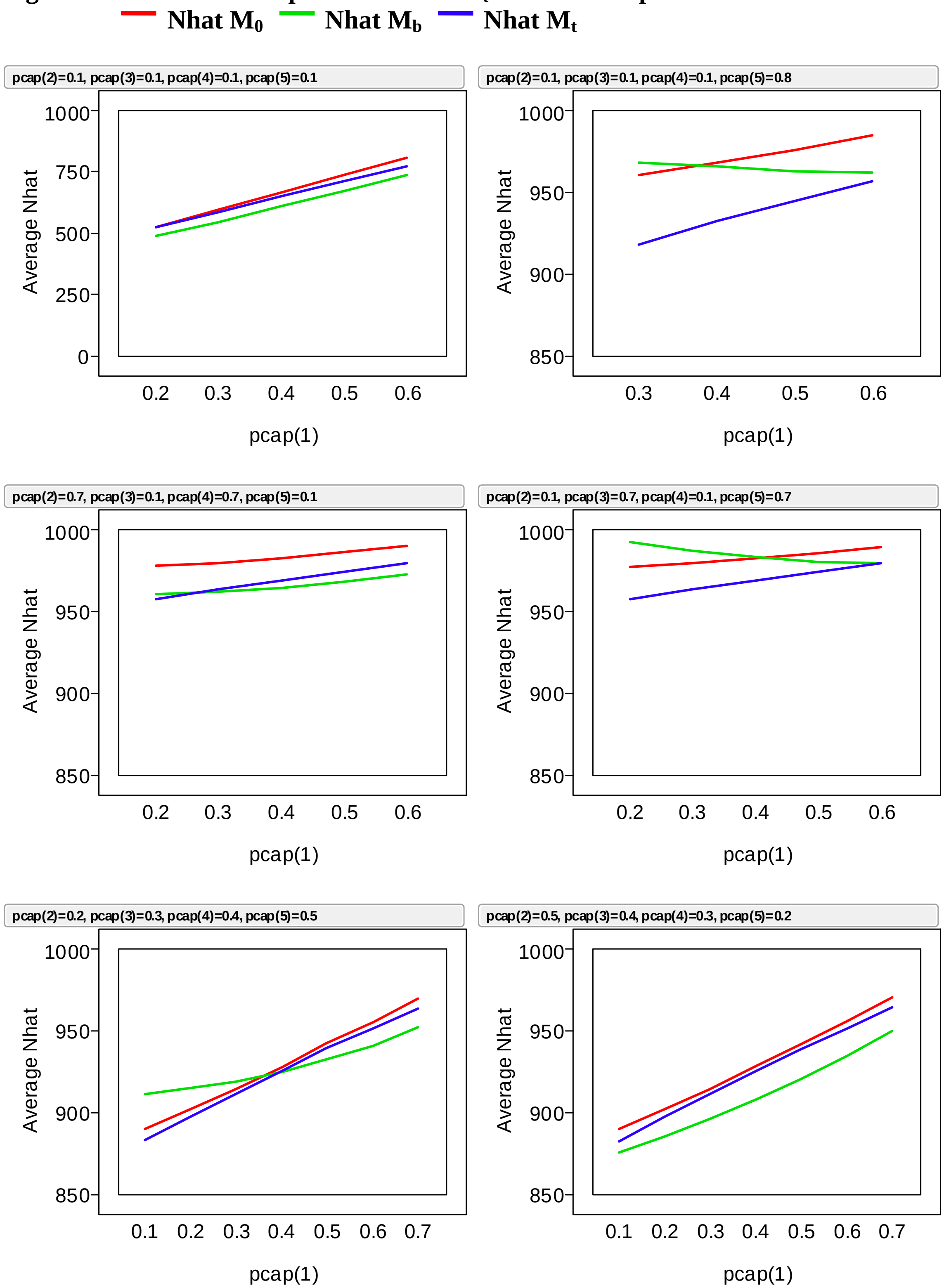
For five capture occasions, graphics were also generated for various combinations of the capture probabilities for the capture occasions. In this case our simulation generated 23,770 lines of data (combinations of capture probabilities); therefore there were a tremendous amount of combinations that could be explored. We chose to only look at a select few. Subsets were created in increments of 0.1 from the simulated data by the probability of capture values on the fifth capture occasion, and graphics were generated from that data. The ones that we selected to examine are seen in Figure 5.3.

In the first panel, we once again see that when the probability of capture is low on all the latter capture occasions, all of the estimators give low estimates of the population, with the $\mathbf{M}_b$ estimator giving the lowest estimates of the three. These patterns are very similar to those that we discussed for three and four capture occasions. As we saw for three and four capture occasions, the behavior appears linear, unlike the marked negative concavity seen for the models discussed in chapters 3 and 4. Also, $\mathbf{M}_0$ and $\mathbf{M}_t$ start out giving very close estimates, but as the capture probability increases for the first capture occasion, they begin to separate. This is similar to what was seen in the first panel of Figure 5.2.

To see how the estimates behaved as the capture probabilities increased on the first capture occasion with a high capture probability on the fifth capture occasion and low capture probabilities on all other occasions, we looked at another combination. On this graphic, we chose to have the capture probability equal to 0.1 on the second, third and fourth capture occasion and to have the capture probability equal to 0.8 on the fifth capture occasion. The estimators for $\mathbf{M}_0$ and $\mathbf{M}_t$ increase relatively linearly as the capture probability on the first capture occasion increases, while the $\mathbf{M}_b$ estimator decreases and shows upward concavity.

We then looked at a combination where the capture probability was high on the second and fourth capture occasions and low on the third and fifth capture occasions. For this combination, we used 0.7 as the capture probability on the second and fourth capture occasion and 0.1 on the third and fifth capture occasion. Each estimate is close to the

Figure 5.3 Selected Examples for Model M_t with Five Capture Occasions



actual population of 1000, with $\mathbf{M}_0$ giving the closest estimate for all values of pcap(1). The $\mathbf{M}_b$ estimator starts out being slightly higher than the $\mathbf{M}_t$ estimator when the capture probability on the first capture occasion is low. As the capture probability on the first capture occasion increases, the $\mathbf{M}_b$ estimator becomes the lowest estimator. There is no real change in the $\mathbf{M}_b$ estimates since there are more marks early; so later capture occasions contribute few unmarks to update N.

Next, we looked at a case where the capture probability was high on the third and fifth capture occasion and low on the second and fourth capture occasion. For this combination, all of the estimators give estimates that are close to the actual population of 1000. $\mathbf{M}_b$ starts out giving the highest estimates for low capture probabilities on the first capture occasion, but since there are more marks late, it begins to decrease as the capture probability increases on the first capture occasion. $\mathbf{M}_0$ gives the highest estimate for the highest capture probability on the first capture occasion. It also appears that $\mathbf{M}_b$ merges with $\mathbf{M}_t$ at the highest capture probability on the first capture occasion.

Next, we chose to look at the behavior of the estimators as the capture probability increased with increasing capture occasions. For this graphic, we chose the capture probability to equal 0.2 on the second capture occasion, 0.3 on the third capture occasion, 0.4 on the fourth capture occasion and 0.5 on the fifth capture occasion. We see that when the capture probability on the first capture occasion is low, the $\mathbf{M}_b$ estimator gives the highest estimate. We also see some positive concavity in the $\mathbf{M}_b$ estimator as the capture probability on the first capture occasion increases. We also see that the $\mathbf{M}_0$ and $\mathbf{M}_t$ estimators steadily increase as the capture probability on the first capture occasion increases. The $\mathbf{M}_b$ estimator then becomes the lowest estimator.

The last combination we chose to examine was one where the capture probabilities decreased as the number of capture occasions increased. For this graphic, we chose the capture probability to equal 0.5 on the second capture occasion, 0.4 on the third capture occasion, 0.3 on the fourth capture occasion, and 0.2 on the fifth occasion. We can clearly see that as the probability of capture increases on the first capture occasion, the $\mathbf{M}_b$ estimator gives the lowest estimates. This can be due to the fact that a lot were marked early, and there were few unmarked left for later. In this case, none of the estimators cross at any point. There also appears to be some slight positive curvature in the $\mathbf{M}_b$ estimator. The $\mathbf{M}_0$ estimator gives the highest estimates and the $\mathbf{M}_t$ estimator lies between the $\mathbf{M}_b$ and $\mathbf{M}_0$ estimates as the capture probability on the first capture occasion increases. The $\mathbf{M}_t$ estimator is closer in its estimates to the $\mathbf{M}_0$ estimator than to the $\mathbf{M}_b$ estimator.

Section 5.4 Bias Estimation in the $\mathbf{M}_t$ Capture Scenario

In Chapters 3 and 4, for models $\mathbf{M}_0$ and $\mathbf{M}_b$, we attempted to model the bias for the estimators. Due to the number of different patterns seen, we did not attempt to model the bias for the $\mathbf{M}_t$ model. We also noted in Chapters 3 and 4 that for low capture probabilities in models $\mathbf{M}_0$ and $\mathbf{M}_b$ the population estimates were significantly biased low. This same pattern holds for the $\mathbf{M}_t$ model. Another observation that we made for model $\mathbf{M}_t$ is that the behavior of the $\mathbf{M}_b$ estimator is erratic. In some cases it is linear, while in others it is concave up. It is also seen crossing the other estimators. It is also seen that the $\mathbf{M}_t$ estimator does not seem to really model its scenario well, whereas the $\mathbf{M}_b$ estimator usually does better.

CHAPTER 6

CONCLUSIONS AND FUTURE PROJECTS

Section 6.1 Conclusions

We began this thesis with the purpose of looking at methods of modeling populations using mark-recapture methods to examine the effects of departures from the assumptions of the estimators. In the introductory chapter we looked at the various models that are available for both open and closed populations. We decided to focus on a closed population. We looked at three of the capture models and their associated estimators, these being models $\mathbf{M}_0$, $\mathbf{M}_b$ and $\mathbf{M}_t$. We wanted to know whether one of these three models performed better than the others in estimating the population, given departures from the assumptions. To try to answer this question, we began looking at the maximum likelihood estimators for each model. We explored these questions through simulated data. We noted that $\mathbf{M}_0$ and $\mathbf{M}_t$ performed equally well for the $\mathbf{M}_0$ and $\mathbf{M}_b$ captures, but $\mathbf{M}_b$ was the better estimator for the $\mathbf{M}_b$ captures. The $\mathbf{M}_t$ estimator for $\mathbf{M}_t$ captures did not perform well. Depending on actual capture probabilities, either of the other two estimators may give better, less biased results.

After looking at the initial simulation results, we became interested in the bias of each estimator as well, since it was clear that in many cases, the population estimates were very low compared to the (known) population used in the simulation. We noticed that the $\mathbf{M}_0$ and $\mathbf{M}_b$ scenarios always gave concave down graphs for the population estimates approaching the true population as an asymptote as the probability of capture increased or as the number of capture occasions increased. We did not notice this behavior pattern in the $\mathbf{M}_t$ scenario. We did some bias modeling for the $\mathbf{M}_0$ and $\mathbf{M}_b$ capture scenarios, which found that probability of capture as a percent (pcappct), number of capture occasions (nocc), number of capture occasions times the probability of capture (nocc*pcap) and a quadratic function in the probability of capture (pcapSq) had good results in terms of r^2, but analysis of the residuals indicates problems with this model.

Section 6.2 Future Projects

There were several things that we noticed in this thesis which we believe could be explored in future projects. First, the bias modeling for models $\mathbf{M}_0$ and $\mathbf{M}_b$ could be explored in more depth. From what we saw in the residuals versus the fitted values graphics, more exploration could be done to obtain a better model (especially one that uses the theoretical asymptote at the actual population level). We saw nonlinear patterns in this model, and the model that we fitted was not of that form. More explorations could also be done for model $\mathbf{M}_t$ to explore its behavior in more depth. We did not explore the bias for this model, since we had such a large amount of simulated data with greatly varied behaviors. We also noted for model $\mathbf{M}_t$ that the $\mathbf{M}_t$ estimator did not behave as well as we thought it should, which could possibly suggest that the derived maximum likelihood estimator may not be the best estimator for this model. Based on these future endeavors, adjusted estimators could be formulated to give less biased results.

REFERENCES

Amstrup, S. C., T. L. McDonald, and B. F. J. Manly. 2005. *Handbook of Capture-Recapture Analysis.* Princeton University Press, New Jersey.

Becker, N. G., and C. C. Heyde. 1990. Estimating population size from multiple recapture experiments. *Stochastic Processes and Their Applications* 36: 77-83.

Begon, Michael. 1979. *Investigating Animal Abundance: capture-recapture for biologists*. Edward Arnold, London.

Blower, J. G., L. M. Cook, and J. A. Bishop. 1981. *Estimating the size of animal populations.* George Allen & Unwin Ltd., London.

Borchers, D. L., S. T. Buckland, and W. Zucchini. 2002. *Statistics for Biology and Health: Estimating Animal Abundance Closed Populations*. Springer-Verlag, London.

Brownie, C., J. E. Hines, and J. D. Nichols. 1986. Constant-parameter capture-recapture models. *Biometrics* 42: 561-574.

Carlson, S. R., L. G. Coggins Jr., and C. O. Swanton. 1998. A simple stratified design for mark-recapture estimation of salmon smolt abundance. *Alaska Fishery Research Bulletin* 5(2): 88-102.

Catchpole, E. A., S. N. Freeman, B. J. T. Morgan, and W. J. Nash. 2001. Abalone I: analyzing mark-recapture-recovery data incorporating growth and delayed recovery. *Biometrics* 57: 469-477.

Chapman, D. G. 1951. Some properties of the hypergeometric distribution with applications to zoological censuses. *University of California Publications in Statistics* 1: 131-160.

Conn, P. B., W. L. Kendall, and M. D. Samuel. 2004. A general model for the analysis of mark-resight, mark-recapture, and band-recovery data under tag loss. *Biometrics* 60: 900-909.

Cormack, R. M. 1964. Estimates of survival from the sightings of marked animals. *Biometrika* 51: 429-438.

Darroch, J. N. 1958. The multiple-recapture census, I: estimation of a closed population. *Biometrika* 45: 343-359.

Darroch, J. N. 1959. The multiple-recapture census, II: estimation when there is immigration or death. *Biometrika* 46: 336-351.

Deroba, J. J., M. J. Hansen, N. A. Nate, and J. M. Hennessy. 2005. Evaluating assumptions of mark-recapture studies for estimating angling exploitation of walleyes in northern Wisconsin lakes. *North American Journal of Fisheries Management* 25: 890-896.

Follner, K., and K. Henle. 2001. Closed mark-recapture models to estimate species richness: an example using data on epigeal spiders. *Journal of Agricultural, Biological, and Environmental Statistics* 6, Number 2: 176-182.

Jarvis, S. N., P. J. Lowe, A. Avery, S. Levene, and R. M. Cormack. 2000. Children are not goldfish – mark/recapture techniques and their application to injury data. *Injury Prevention*, 6: 46-50.

Jolly, G. M. 1982. Mark-recapture models with parameters constant in time. *Biometrics* 38: 301-321.

Lloyd, C. J., and P. S. F. Yip. 1991. A unification of inference from capture-recapture studies through martingale estimating functions. *Estimating Functions*, V. P. Godambe, editor. Oxford University Press, Oxford, England.

Manly, B. F. J., L. L. McDonald, and T. L. McDonald. 1999. The robustness of mark-recapture methods: a case study for the northern spotted owl. *Journal of Agricultural, Biological, and Environmental Statistics* 4, Number 2: 78-101.

O'Brien, S., B. Robert, and H. Tiandry. 2005. Consequences of violating the recapture duration assumption of mark-recapture models: a test using simulated and empirical data from an endangered tortoise population. *Journal of Applied Ecology*, 42: 1096-1104.

Otis, D. L., K. P. Burnham, G. C. White, and D. R. Anderson. 1978. Statistical inference from capture data on closed animal populations. *Wildlife Monographs* 62: 1-135.

Parmenter, C. A., T. L. Yates, R. R. Parmenter, J. N. Mills, J. E. Childs, M. L. Campbell, J. L. Dunnum, and J. Milner. 1998. Small mammal survival and trapability in mark-recapture monitoring programs for hantavirus. *Journal of Wildlife Diseases* 34(1): 1-12.

Pollock, K. H., J. E. Hines, and J. D. Nichols. 1985. Goodness-of-fit tests for open capture-recapture models. *Biometrics* 41: 399–410.

Pollock, K.H., J. D. Nichols, C. Brownie, and J. E. Hines. 1990. Statistical inference for capture-recapture experiments. *Wildlife Monographs* 107: 1-97.

Ricker, W. E. 1975. Computation and Interpretation of Biological Statistics of Fish Populations. *Bulletin of the Fisheries Research Board of Canada 191*.

Rosenberger, A. E., and J. B. Dunham. 2005. Validation of abundance estimates from mark-recapture and removal techniques for rainbow trout captured by electrofishing in small streams. *North American Journal of Fisheries Management* 25: 1395-1410.

Schnabel, Z. E. 1938. The estimation of the total fish population of a lake. *American Mathematical Monthly* 45: 348-352.

Seber, G. A. F. 2001. Some new directions in estimating animal population parameters. *Journal of Agricultural, Biological, and Environmental Statistics* 6, Number 2: 140-151.

Skalski, J. R., K. E. Ryding, and J. J. Millspaugh. 2005. *Wildlife Demography: Analysis of Sex, Age, and Count Data.* Elsevier Academic Press, Massachusetts, California, London.

White, G. C., D. R. Anderson, K. P. Burnham, and D. L. Otis. 1982. *Capture-recapture and removal methods for sampling closed populations.* Los Alamos National Lab, LA-8787-NERP, Los Alamos, NM.

Yip, P. S. F, E. C. Y. Wan, and K. S. Chan. 2001. A unified approach for estimating population size in capture-recapture studies with arbitrary removals. *Journal of Agricultural, Biological, and Environmental Statistics* 6, Number 2: 183-194.

Yip, P. S. F. 1993. Statistical inference procedure for a hypergeometric model for capture-recapture experiments. *Applied Mathematics and Computation* 58: 225-240.

Zippin, C. 1956. An evaluation of the removal method of estimating animal populations. *Biometrics* 12: 163-189.

APPENDICES

APPENDIX A

FORTRAN PROGRAM FOR MODEL M_0

```
c      program M0 simulation
c
c      program to estimate population parameters based on actual data.
c      number of capture occasions and marks/unmarks are input for each
c      simulation estimates standard deviations of parameters. This
c        version is for model M0 - all capture probabilities are the
c        same each occasion for all animals. Two parameters to estimate:
c        N and p.
c
c
       character*10 outfile
       integer*4 n, cmarks, unmarks, oldpop, totcap, nsubj(25), cm(25),
     1   u(25), bigmdot, sumnb, nsqb, Nmin, Nmax, Nmint, Nmaxt, nsqt
       parameter (ndim=2,mp=3,np=2)
       real*4 xguess(2), x(2), Nave, Nsq, newp(2), maxlik, SDN,
     1  xscale(2), p(mp,np), y(mp), Nlo, Nhi, sump, sumn, snp, likeli,
     2  phat, pcap, phatt(25),naveb, phatb, Navet, pmint(25), pmaxt(25),
     3 sumpt(25), psqt(25), sdpt(25), plowt(25), phit(25), pavet(25),
     4  plow, phi
        idum = -1
c
c      Accept parameters for the simulation
c
       print 4
4      format (' Enter the Total Population')
       read(*,5) in
5      format(i4)
       print 8
8      format(' Input the number of repetitions for simulation i4')
       read(*,5) nrep
       print 2
2      format (' Enter random seed - large odd - i7')
        read (*,6) idum
6       format (i7)
        print 9
9       format (' Enter output file name (10 characters)')
        read(*,11) outfile
11      format (a10)
       open (unit=10,status='unknown',file=outfile)
        do 850 nocc=3,10
        do 860 ipcap=1,8
        pcap = ipcap/10.
c
c       Initialize Variables
c
        nave = 0
        pave = 0.
        naveb = 0
        caveb = 0.
        navet = 0
        cmaxb = 0.
        cminb = 1.
        chi = 0.
        sumpb = 0.
        paveb = 0.
```

```
        psqb = 0.
        pminb = 1.
        pmaxb = 0.
        csqb = 0.
        nsqb = 0.
        naveb = 0.
        sumnb = 0.
        nminb = 8000.
        nmaxb = 0.
        Nmin = 8000
        Nmax = 0
        Pmin = 1.
        Pmax = 0.
        Nmint = 8000
        Nmaxt = 0
        do 145 i = 1,25
        sumpt(i) = 0.
        pavet(i) = 0.
        psqt(i) = 0.
        pmint(i) = 1.
145      pmaxt(i) = 0.
        sumnt = 0
        Navet = 0.
        nsqt = 0
c
c      simulate mark/recapture
c
       do 50 ijk = 1,nrep
       do 100, j=1,nocc
       if (j .eq. 1) unmarks = in
c      Capture Portion
       if (j .eq. 1) then
       mdot = 0
       bigmdot = 0
        acm = bnldev(pcap, unmarks, idum)
       cmarks = nint(acm)
       totcap = cmarks
       cm(1) = 0
       u(1)  = cmarks
       unmarks = unmarks - cmarks
       else
       bigmdot = bigmdot + cmarks
       acm = bnldev(pcap,cmarks,idum)
       cm(j) = nint(acm)
       mdot = mdot + cm(j)
       acu = bnldev(pcap,unmarks,idum)
       u(j)  = nint(acu)
       cmarks = cmarks + u(j)
       unmarks = unmarks - u(j)
       totcap = totcap + cm(j) + u(j)
       endif
100    continue
c
c      End of simulation portion.  Estimate parameters for model M0.
c
        nlim = 2*in
        maxlik = 0
        do 200 nhat = cmarks, nlim
        likeli = alnfact(nhat,cmarks) +
     1      totcap*log(real(totcap)) +
     2      (nocc*nhat - totcap)*log(real(nocc*nhat - totcap)) -
     3      nocc*nhat*log(real(nocc*nhat))
        if (likeli .ge. maxlik) then
```

```
            maxlik = likeli
            go to 200
         else
            go to 250
         endif
200      continue
250      continue
        nhat = nhat - 1
        phat = real(totcap)/real(nocc*nhat)
c
c           Now, compute estimates for model Mt
c
         nlim = 2*in
         maxlik = 0
         do 210 nhatt = cmarks, nlim
         sum1 = 0
         sum2 = 0
         do 220 j=1, nocc
         anj = real(u(j) + cm(j))
         sum1 = sum1 + anj*log(anj)
         sum2 = sum2 + (nhatt - anj)*log(real(nhatt)-anj)
220      continue
         likeli = alnfact(nhatt,cmarks) + sum1 + sum2 -
     1        nocc*nhatt*log(real(nhatt))
         if (likeli .ge. maxlik) then
            maxlik = likeli
            go to 210
         else
            go to 240
         endif
210      continue
240      continue
        nhatt = nhatt - 1
        do 222  j=1, nocc
222     phatt(j) = (u(j)+cm(j))/real(nhatt)

c
c           Now, compute estimates for model Mb
c
         nlim = 2*in
         maxlik = 0
         do 235 nhatb = cmarks, nlim
         likeli = alnfact(nhatb,cmarks) + cmarks*log(real(cmarks)) +
     1   (nocc*nhatb-bigmdot-cmarks)*log(real(nocc*nhatb-bigmdot-cmarks))
     2    -(nocc*nhatb-bigmdot)*log(real(nocc*nhatb-bigmdot))
         if (likeli .ge. maxlik) then
            maxlik = likeli
            go to 235
         else
            go to 245
         endif
235      continue
245      continue
        nhatb = nhatb - 1
        phatb = real(cmarks)/real(nocc*nhatb-bigmdot)
        chat = real(mdot)/real(bigmdot)
c
c       accumulate summary estimates for M0
c
35      format(' M0 N hat = ', i5, ' phat = ', f6.3)
c        write (10,900) phat, nhat
900     format ('M0 estimates P = ',f7.5,' N = ',i5)
        if (nhat .le. Nmin) Nmin = nhat
```

```
      if (nhat .ge. Nmax) Nmax = nhat
      sumn = sumn + nhat
      nave = nave + nhat/float(nrep)
      nsq  = nsq + nhat**2
      if (phat .le. Pmin) Pmin = phat
      if (phat .ge. Pmax) Pmax = phat
      sump = sump + phat
      Pave = Pave + phat/float(nrep)
      Psq  = Psq + phat**2
      Snp = snp + phat*nhat
c
c      accumulate summary estimates for model Mt
c
45      format(' N hatt = ', i5, ' phatt = ', 11(f6.3, 1x))
c       write (10,45) phatt, nhatt
      if (nhatt .le. Nmint) Nmint = nhatt
      if (nhatt .ge. Nmaxt) Nmaxt = nhatt
      sumnt = sumnt + nhatt
      Navet = Navet + nhatt/float(nrep)
      Nsqt  = Nsqt + nhatt**2
       do 29 j = 1, nocc
      if (phatt(j) .le. pmint(j)) pmint(j) = phatt(j)
      if (phatt(j) .ge. pmaxt(j)) pmaxt(j) = phatt(j)
      sumpt(j) = sumpt(j) + phatt(j)
      pavet(j) = pavet(j) + phatt(j)/float(nrep)
29    psqt(j)  = psqt(j) + phatt(j)**2
c     Snpt = snpt + phatt*nhatt
c
c      accumulate summary estimates for model Mb
c
55     format(' N hatb = ', i5, ' phatb = ',f6.3, ' chat = ',f6.3)
c       write (10,55) nhatb, phatb, chat
      if (nhatb .le. nminb) nminb = nhatb
      if (nhatb .ge. nmaxb) nmaxb = nhatb
      sumnb = sumnb + nhatb
      naveb = naveb + real(nhatb)/real(nrep)
      nsqb  = nsqb + nhatb**2
      if (phatb .le. pminb) pminb = phatb
      if (phatb .ge. pmaxb) pmaxb = phatb
      sumpb = sumpb + phatb
      paveb = paveb + phatb/float(nrep)
      psqb  = psqb + phatb**2
      Snpb = snpb + phatb*real(nhatb)
       if (chat .le. Cminb) Cminb = chat
      if (chat .ge. Cmaxb) Cmaxb = chat
      sumc = sumc + chat
      caveb = caveb + chat/float(nrep)
      csqb  = csqb + chat**2
      sncb = sncb + chat*nhatb

50     continue
            arep = float(nrep)
c
c       Wrap up for M0
c
      SDN = sqrt((Nsq - arep*Nave**2)/(arep - 1.0))
      SdP = sqrt((Psq - arep *Pave**2)/(arep - 1.0))
      sdn0 = Nave*((1-pave)**(-nocc)-nocc*(1-pave)**(-1)+nocc-1)**(-1)
      Nlo = nave - 1.96*SDN
      if (Nlo .le. 0.0) Nlo = 0.0
      Nhi = nave + 1.96*SDN
      plow   = pave - 1.96*SdP
      if (plow .le. 0.0) plow = 0.0
```

```
      phi = pave + 1.96*SdP
      cornp = (snp - (sumn*sump)/arep)/(
     1  (nsq -sumn**2/arep)*(psq-sump**2/arep))**.5
c
c       Do wrap up for Mt and Mb
c
c       Wrap up for Mt
c
      SdNt = ((real(Nsqt) - arep*Navet**2)/(arep - 1.0))**.5
      Nlo = navet - 1.96*SDNt
      if (Nlo .le. 0.0) Nlo = 0.0
      Nhi = navet + 1.96*SDNt
      phatprod = 1.
        pinvsum = 0.
      do 49 j = 1,nocc
        sdpt(j) = ((psqt(j) - arep *pavet(j)**2)/(arep - 1.0))**.5
      plowt(j) = pavet(j) - 1.96*sdpt(j)
      if (plowt(j) .le. 0.0) plowt(j) = 0.0
      phit(j) = pavet(j) + 1.96*sdpt(j)
      phatprod = (1 - pavet(j))*phatprod
      pinvsum = pinvsum + (1-pavet(j))**(-1)
49      continue
        sdnt = navet*(1/phatprod + nocc - 1 - pinvsum)**(-1)
c
c       Wrap up for Mb
c
      SDNb = ((nsqb - sumnb**2/arep)/(arep - 1.0))**.5
      if (SDN .le. 0.000000001) SDN = 0.
      print *, SDN
      SdP = ((psqb - arep*paveb**2)/(arep - 1.0))**.5
      SdC = ((csqb - arep*caveb**2)/(arep - 1.0))**.5
        sdnb =(naveb*(1-paveb)**nocc*(1-(1-paveb)**nocc))/(((1-(1-paveb)
     1    **nocc)**2)-nocc**2*paveb**2*(1-paveb)**(nocc-1))
      Nlo = naveb - 1.96*SDN
      if (Nlo .le. 0.0) Nlo = 0.0
      Nhi = naveb + 1.96*SDN
      plow = paveb - 1.96*SdP
      if (plow .le. 0.0) plow = 0.0
      phi = paveb + 1.96*SdP
      cornp = (1/(real(nocc)-1))*(snpb - (sumnb*sumpb)/arep)/(
     1  (nsqb -sumnb**2/arep)*(psqb-sumpb**2/arep))**.5
        clow   = caveb - 1.96*SdC
      if (clow .le. 0.0) clow = 0.0
      chi = caveb + 1.96*SdC
      cornc = (1/(real(nocc)-1))*(sncb - (sumnb*sumc)/arep)/(
     1  (nsqb - sumnb**2/arep)*(csqb-sumc**2/arep))**.5
        write(10,840) nocc, pcap, nave, sdn0, pave, naveb, sdnb, paveb,
     1    caveb, navet, sdnt, (pavet(j), j=1,nocc)
840     format(i2,1x,f4.2, 19(1x,f8.3))
860   continue
850   continue
        print 101
101     format(' Finished')
      close(unit=10)
      end

        Function alnfact(nhat,cmarks)
c
c       Function to compute ln((Nhat)!/(Nhat-cmarks)!)
c
        integer cmarks
        afact=0.0
        bfact = 0.0
```

```
        do 1 j = 1,nhat
        afact = afact + log(real(j))
1       continue
        ik = nhat - cmarks
        Do 2 j = 1,ik
2       bfact = bfact + log(real(ik))
        alnfact = afact - bfact
        return
        end
c
       FUNCTION ran1(idum)
c
c       Function used by Binomal random variate function
c
       INTEGER idum,IA,IM,IQ,IR,NTAB,NDIV
       REAL ran1,AM,EPS,RNMX
       PARAMETER (IA=16807,IM=2147483647,AM=1./IM,IQ=127773,IR=2836,
     1  NTAB=32,NDIV=1+(IM-1)/NTAB,EPS=1.2e-7,RNMX=1.-EPS)
       INTEGER j,k,iv(NTAB),iy
       SAVE iv,iy
       DATA iv /NTAB*0/, iy /0/
c      if (idum.le.0.or.iy.eq.0) then
c      idum=max(-idum,1)
       do 11 j=NTAB+8,1,-1
       k=idum/IQ
       idum=IA*(idum-k*IQ)-IR*k
       if (idum.lt.0) idum=idum+IM
       if (j.le.NTAB) iv(j)=idum
11     continue
       iy=iv(1)
c      endif
       k=idum/IQ
       idum=IA*(idum-k*IQ)-IR*k
       if (idum.lt.0) idum=idum+IM
       j=1+iy/NDIV
       iy=iv(j)
       iv(j)=idum
       ran1=min(AM*iy,RNMX)
       return
       END
C        (C) Copr. 1986-92 Numerical Recipes Software 0Q-815='.

       FUNCTION bnldev(pp,n,idum)
C       Subroutine to generate binomial rancom observations for capture
C         simulation
C
       INTEGER idum,n
       REAL*4 bnldev,pp,PI
C    USES gammln,ran1
       PARAMETER (PI=3.141592654)
       INTEGER j,nold
       REAL am,em,en,g,oldg,p,pc,pclog,plog,pold,sq,t,y,gammln,ran1
       SAVE nold,pold,pc,plog,pclog,en,oldg
       DATA nold /-1/, pold /-1./
       if(pp.le.0.5)then
       p=pp
       else
       p=1.-pp
       endif
       am=n*p
       if (n.lt.25)then
       bnldev=0.
       do 11 j=1,n
```

```
      if(ran1(idum).lt.p)bnldev=bnldev+1.
11    continue
      else if (am.lt.1.) then
      g=exp(-am)
      t=1.
      do 12 j=0,n
      t=t*ran1(idum)
      if (t.lt.g) goto 1
12    continue
      j=n
1     bnldev=j
      else
      if (n.ne.nold) then
      en=n
      oldg=gammln(en+1.)
      nold=n
      endif
      if (p.ne.pold) then
      pc=1.-p
      plog=log(p)
      pclog=log(pc)
      pold=p
      endif
      sq=sqrt(2.*am*pc)
2     y=tan(PI*ran1(idum))
      em=sq*y+am
      if (em.lt.0..or.em.ge.en+1.) goto 2
      em=int(em)
      t=1.2*sq*(1.+y**2)*exp(oldg-gammln(em+1.)-gammln(en-em+1.)
     1    +em*plog+(en-em)*pclog)
      if (ran1(idum).gt.t) goto 2
      bnldev=em
      endif
      if (p.ne.pp) bnldev=n-bnldev
      return
      END
C  (C) Copr. 1986-92 Numerical Recipes Software 0Q-815='.
      FUNCTION gammln(xx)
c
c       Function used by Amoeba routine
c
      REAL gammln,xx
      INTEGER j
      DOUBLE PRECISION ser,stp,tmp,x,y,cof(6)
      SAVE cof,stp
      DATA cof,stp/76.18009172947146d0,-86.50532032941677d0,
     1  24.01409824083091d0,-1.231739572450155d0,.1208650973866179d-2,
     2  -.5395239384953d-5,2.5066282746310005d0/
      x=xx
      y=x
      tmp=x+5.5d0
      tmp=(x+0.5d0)*log(tmp)-tmp
      ser=1.000000000190015d0
      do 11 j=1,6
      y=y+1.d0
      ser=ser+cof(j)/y
11    continue
      gammln=tmp+log(stp*ser/x)
      return
      END
C  (C) Copr. 1986-92 Numerical Recipes Software 0Q-815='.
```

APPENDIX B

FORTAN PROGRAM FOR MODEL M_b

```
c      program Mb simulation
c
c      program to estimate population parameters based on actual data.
c      number of capture occasions and marks/unmarks are input for each
c      simulation estimates standard deviations of parameters. This
c        version is for model M0 - all capture probabilities are the
c        same each occasion for all animals. Two parameters to estimate:
c        N and p.
c
c
       character*10 outfile
       integer*4 n, cmarks, unmarks, oldpop, totcap, nsubj(25), cm(25),
     1   u(25), bigmdot, sumnb, nsqb, Nmin, Nmax, Nmint, Nmaxt, nsqt
c
c      np = number of parameters to estimate
c      mp = number of parameters + 1
c      matrix p holds initial values and is used in the estimating
c      routine
c
       parameter (ndim=2,mp=3,np=2)
       real*4 xguess(2), x(2), Nave, Nsq, newp(2), maxlik,
     1  xscale(2), p(mp,np), y(mp), Nlo, Nhi, sump, sumn, snp, likeli,
     2  phat, pcap, phatt(25),naveb, phatb, Navet, pmint(25), pmaxt(25),
     3 sumpt(25), psqt(25), sdpt(25), plowt(25), phit(25), pavet(25),
     4  plow, phi
c
c      Accept parameters for the simulation
c
       print 4
4      format (' Enter the Total Population')
       read(*,5) in
5      format(i4)
       print 8
8      format(' Input the number of repetitions for simulation i4')
       read(*,5) nrep
       print 2
2      format (' Enter random seed - large odd i7')
        read(*,6) idum
6       format (i7)
        print 9
9       format (' Enter output file name (10 characters)')
        read(*,11) outfile
11      format (a10)
       open (unit=10,status='unknown',file=outfile)
        do 850 nocc=3,10
        do 860 ipcap=1,8
        do 870 idiff = -3, 3
        ircap = ipcap+idiff
        if(ircap.lt.1) go to 870
        if(ircap.gt.8) go to 870
        if(ircap.eq.ipcap) go to 870
        rcap=ircap/10.
        pcap = ipcap/10.
c
c       Initialize Variables
c
        nave = 0
        pave = 0.
```

```
          naveb = 0
          caveb = 0.
          navet = 0
          cmaxb = 0.
          cminb = 1.
          chi = 0.
          sumpb = 0.
          paveb = 0.
          psqb = 0.
          pminb = 1.
          pmaxb = 0.
          csqb = 0.
          nsqb = 0.
          naveb = 0.
          sumnb = 0.
          nminb = 8000.
          nmaxb = 0.
          Nmin = 8000
          Nmax = 0
          Pmin = 1.
          Pmax = 0.
          Nmint = 8000
          Nmaxt = 0
          do 145 i = 1,25
          sumpt(i) = 0.
          pavet(i) = 0.
          psqt(i) = 0.
          pmint(i) = 1.
145        pmaxt(i) = 0.
          sumnt = 0
          Navet = 0.
          nsqt = 0
c
c      simulate mark/recapture
c
      do 50 ijk = 1,nrep
      do 100, j=1,nocc
      if (j .eq. 1) unmarks = in
c     Capture Portion
      if (j .eq. 1) then
      mdot = 0
      bigmdot = 0
        acm = bnldev(pcap, unmarks, idum)
      cmarks = nint(acm)
      totcap = cmarks
      cm(1) = 0
      u(1)  = cmarks
      unmarks = unmarks - cmarks
      else
      bigmdot = bigmdot + cmarks
      acm = bnldev(rcap,cmarks,idum)
      cm(j) = nint(acm)
      mdot = mdot + cm(j)
      acu = bnldev(pcap,unmarks,idum)
      u(j)  = nint(acu)
      cmarks = cmarks + u(j)
      unmarks = unmarks - u(j)
      totcap = totcap + cm(j) + u(j)
      endif
100   continue
c       print *, totcap, cmarks
c     print 901, (u(i),i=1,10)
c     print 901, (cm(i), i=1,10)
```

```
901    format (1x,10(1x,i4.0))
c
c      End of simulation portion.  Estimate parameters for model M0.
c
        nlim = 2*in
        maxlik = 0
        do 200 nhat = cmarks, nlim
        likeli = alnfact(nhat,cmarks) +
     1      totcap*log(real(totcap)) +
     2      (nocc*nhat - totcap)*log(real(nocc*nhat - totcap)) -
     3      nocc*nhat*log(real(nocc*nhat))
        if (likeli .ge. maxlik) then
           maxlik = likeli
           go to 200
        else
           go to 250
        endif
200     continue
250     continue
       nhat = nhat - 1
       phat = real(totcap)/real(nocc*nhat)
c
c          Now, compute estimates for model Mt
c
        nlim = 2*in
        maxlik = 0
        do 210 nhatt = cmarks, nlim
        sum1 = 0
        sum2 = 0
        do 220 j=1, nocc
        anj = real(u(j) + cm(j))
        sum1 = sum1 + anj*log(anj)
        sum2 = sum2 + (nhatt - anj)*log(real(nhatt)-anj)
220     continue
        likeli = alnfact(nhatt,cmarks) + sum1 + sum2 -
     1      nocc*nhatt*log(real(nhatt))
        if (likeli .ge. maxlik) then
           maxlik = likeli
           go to 210
        else
           go to 240
        endif
210     continue
240     continue
       nhatt = nhatt - 1
       do 222  j=1, nocc
222    phatt(j) = (u(j)+cm(j))/real(nhatt)

c
c          Now, compute estimates for model Mb
c
        nlim = 2*in
        maxlik = 0
        do 235 nhatb = cmarks, nlim
        likeli = alnfact(nhatb,cmarks) + cmarks*log(real(cmarks)) +
     1  (nocc*nhatb-bigmdot-cmarks)*log(real(nocc*nhatb-bigmdot-cmarks))
     2   -(nocc*nhatb-bigmdot)*log(real(nocc*nhatb-bigmdot))
        if (likeli .ge. maxlik) then
           maxlik = likeli
           go to 235
        else
           go to 245
        endif
```

```
235     continue
245     continue
        nhatb = nhatb - 1
        phatb = real(cmarks)/real(nocc*nhatb-bigmdot)
        chat = real(mdot)/real(bigmdot)
c
c       print estimates for M0
c
35      format(' M0 N hat = ', i5, ' phat = ', f6.3)
c              write (10,900) phat, nhat
900     format ('M0 estimates P = ',f7.5,' N = ',i5)
        if (nhat .le. Nmin) Nmin = nhat
        if (nhat .ge. Nmax) Nmax = nhat
        sumn = sumn + nhat
        nave = nave + nhat/float(nrep)
        nsq  = nsq + nhat**2
        if (phat .le. Pmin) Pmin = phat
        if (phat .ge. Pmax) Pmax = phat
        sump = sump + phat
        Pave = Pave + phat/float(nrep)
        Psq  = Psq + phat**2
        Snp = snp + phat*nhat
c
c       print estimates for model Mt
c
         print 45, nhatt, (phatt(i), i=1,nocc)
45       format(' N hatt = ', i5, ' phatt = ', 11(f6.3, 1x))
c              write (10,45) phatt, nhatt
        if (nhatt .le. Nmint) Nmint = nhatt
        if (nhatt .ge. Nmaxt) Nmaxt = nhatt
        sumnt = sumnt + nhatt
        Navet = Navet + nhatt/float(nrep)
        Nsqt  = Nsqt + nhatt**2
         do 29 j = 1, nocc
        if (phatt(j) .le. pmint(j)) pmint(j) = phatt(j)
        if (phatt(j) .ge. pmaxt(j)) pmaxt(j) = phatt(j)
        sumpt(j) = sumpt(j) + phatt(j)
        pavet(j) = pavet(j) + phatt(j)/float(nrep)
29      psqt(j)  = psqt(j) + phatt(j)**2
c       Snpt = snpt + phatt*nhatt
c
c       print estimates for model Mb
c
c         print 55, nhatb, phatb, chat
55      format(' N hatb = ', i5, ' phatb = ',f6.3, ' chat = ',f6.3)
c              write (10,55) nhatb, phatb, chat
        if (nhatb .le. nminb) nminb = nhatb
        if (nhatb .ge. nmaxb) nmaxb = nhatb
        sumnb = sumnb + nhatb
        naveb = naveb + real(nhatb)/real(nrep)
        nsqb  = nsqb + nhatb**2
        if (phatb .le. pminb) pminb = phatb
        if (phatb .ge. pmaxb) pmaxb = phatb
        sumpb = sumpb + phatb
        paveb = paveb + phatb/float(nrep)
        psqb  = psqb + phatb**2
        Snpb = snpb + phatb*real(nhatb)
         if (chat .le. Cminb) Cminb = chat
        if (chat .ge. Cmaxb) Cmaxb = chat
        sumc = sumc + chat
        caveb = caveb + chat/float(nrep)
        csqb  = csqb + chat**2
        sncb = sncb + chat*nhatb
```

```
50     continue
              arep = float(nrep)
c
c       Wrap up for M0
c
c        write(10,503)
503     format(///, '                     Summary for Simulation', //)
       SDN = sqrt((Nsq - arep*Nave**2)/(arep - 1.0))
       SdP = sqrt((Psq - arep *Pave**2)/(arep - 1.0))
       Nlo = nave - 1.96*SDN
       if (Nlo .le. 0.0) Nlo = 0.0
       Nhi = nave + 1.96*SDN
       plow   = pave - 1.96*SdP
       if (plow .le. 0.0) plow = 0.0
       phi = pave + 1.96*SdP
       cornp = (snp - (sumn*sump)/arep)/(
     1  (nsq -sumn**2/arep)*(psq-sump**2/arep))**.5
c      print 31, nave, SdN, Nlo, Nhi
31     format( 'M0 Nhat= ',f8.3, ' sd(N) = ',f8.4,' Low 95% = ',f7.3,
     1 ' Hi 95% = ', f8.3)
c      print 33, Pave, SdP, Plow, phi
33     format( 'M0 P hat= ',f7.3, ' sd(P) = ',f8.4,' Low 95% = ',f7.3,
     1  ' Hi 95% = ', f10.3)
c      print 34, cornp
34     format ('M0 corr n,p = ',f9.4)
c      write (10,31) nave, SdN, Nlo, Nhi
c      write (10,52) Nmin, Nmax
c      write (10,33) Pave, SdP, Plow, phi
c      write (10,58) Pmin, Pmax
c
c       Do wrap up for Mt and Mb
c
c       Wrap up for Mt
c
       SdNt = ((Nsqt - arep*Navet**2)/(arep - 1.0))**.5
       Nlo = navet - 1.96*SDNt
       if (Nlo .le. 0.0) Nlo = 0.0
       Nhi = navet + 1.96*SDNt
c      print 41, nave, SdN, Nlo, Nhi
41     format( 'Mt Nhat= ',f8.3, ' sd(N) = ',f8.4,' Low 95% = ',f7.3,
     1 ' Hi 95% = ', f8.3)
c      write (10,41) navet, SdNt, Nlo, Nhi
c      write(10,52) Nmint, Nmaxt
       do 49 j = 1,nocc
         sdpt(j) = ((psqt(j) - arep *pavet(j)**2)/(arep - 1.0))**.5
       plowt(j) = pavet(j) - 1.96*sdpt(j)
       if (plowt(j) .le. 0.0) plowt(j) = 0.0
       phit(j) = pavet(j) + 1.96*sdpt(j)
c      cornp = (snp - (sumn*sump)/arep)/(
c    1  (nsq -sumn**2/arep)*(psq-sump**2/arep))**.5
c      print 43, j, pavet(j), sdpt(j), plowt(j), phit(j)
43     format( 'Mt P hat(',i2,')= ',f9.4, ' sd(P) = ',f9.4,
     1  ' Low 95% = ',f9.4,' Hi 95% = ', f9.4)
c       print 58, pmint(j), pmaxt(j)
c      print 44, cornp
44     format ('Mt corr n,p = ',f9.4)
c      write (10,43) j, pavet(j), sdpt(j), plowt(j), phit(j)
c      write (10,58) pmint(j), pmaxt(j)
49      continue
c
c       Wrap up for Mb
c
```

```
      SDN = ((nsqb - arep*naveb**2)/(arep - 1.0))**.5
      SdP = ((psqb - arep*paveb**2)/(arep - 1.0))**.5
      SdC = ((csqb - arep*caveb**2)/(arep - 1.0))**.5
      Nlo = naveb - 1.96*SDN
      if (Nlo .le. 0.0) Nlo = 0.0
      Nhi = naveb + 1.96*SDN
      plow = paveb - 1.96*SdP
      if (plow .le. 0.0) plow = 0.0
      phi = paveb + 1.96*SdP
      cornp = (snpb - (sumnb*sumpb)/arep)/(
     1  (nsqb -sumnb**2/arep)*(psqb-sumpb**2/arep))**.5
       clow   = caveb - 1.96*SdC
      if (clow .le. 0.0) clow = 0.0
      chi = caveb + 1.96*SdC
      cornc = (sncb - (sumnb*sumc)/arep)/(
     1  (nsqb -sumnb**2/arep)*(csqb-sumc**2/arep))**.5
c     print 51, naveb, SdN, Nlo, Nhi
51    format( 'Mb N hat= ',f8.3, ' sd(N) = ',f8.4,' Low 95% = ',f7.3,
     1 ' Hi 95% = ', f8.3)
c     print 53, Paveb, SdP, Plow, phi
53    format( 'Mb P hat= ',f9.4, ' sd(P) = ',f8.4,' Low 95% = ',f7.3,
     1  ' Hi 95% = ', f10.3)
c       print 56, caveb, SdC, clow, chi
56    format( 'Mb C hat= ',f9.4, ' sd(C) = ',f8.4,' Low 95% = ',f7.3,
     1  ' Hi 95% = ', f10.3)
c     write (10,51) naveb, SdN, Nlo, Nhi
c     write (10,52) nminb, nmaxb
52     format (' Min = ', i5, ' Max = ', i5)
c     write (10,53) Paveb, SdP, plow, phi
c     write (10,58) pminb, pmaxb
58     format(' Min = ', f9.5, ' Max = ', f9.5)
c     write (10,56) caveb, SdC, clow, chi
c     write (10, 58) cminb, cmaxb
c     write (10,57) cornp, cornc
57     format( 'Mb corr (n,p) = ', f8.5, ' corr(n,c) = ', f8.5)
c
       write(10,840) nocc, pcap, rcap, nave, pave, naveb, paveb, caveb,
     1   navet,(pavet(j), j=1,nocc)
840    format(i2,1x,f4.2,1x,f4.2, 16(1x,f8.3))
870    continue
860   continue
850   continue
      close(unit=10,disp='keep')
      end

       Function alnfact(nhat,cmarks)
c
c      Function to compute ln((Nhat)!/(Nhat-cmarks)!)
c
       integer cmarks
       afact=0.0
       bfact = 0.0
       do 1 j = 1,nhat
       afact = afact + log(real(j))
1      continue
       ik = nhat - cmarks
       Do 2 j = 1,ik
2      bfact = bfact + log(real(ik))
       alnfact = afact - bfact
       return
       end
c
      FUNCTION ran1(idum)
```

```
c
c        Function used by Binomal random variate function
c
       INTEGER idum,IA,IM,IQ,IR,NTAB,NDIV
       REAL ran1,AM,EPS,RNMX
       PARAMETER (IA=16807,IM=2147483647,AM=1./IM,IQ=127773,IR=2836,
     1  NTAB=32,NDIV=1+(IM-1)/NTAB,EPS=1.2e-7,RNMX=1.-EPS)
       INTEGER j,k,iv(NTAB),iy
       SAVE iv,iy
       DATA iv /NTAB*0/, iy /0/
c      if (idum.le.0.or.iy.eq.0) then
c      idum=max(-idum,1)
       do 11 j=NTAB+8,1,-1
       k=idum/IQ
       idum=IA*(idum-k*IQ)-IR*k
       if (idum.lt.0) idum=idum+IM
       if (j.le.NTAB) iv(j)=idum
11     continue
       iy=iv(1)
c      endif
       k=idum/IQ
       idum=IA*(idum-k*IQ)-IR*k
       if (idum.lt.0) idum=idum+IM
       j=1+iy/NDIV
       iy=iv(j)
       iv(j)=idum
       ran1=min(AM*iy,RNMX)
       return
       END
C        (C) Copr. 1986-92 Numerical Recipes Software 0Q-815='.

       FUNCTION bnldev(pp,n,idum)
C        Subroutine to generate binomial rancom observations for capture
C          simulation
C
       INTEGER idum,n
       REAL*4 bnldev,pp,PI
C     USES gammln,ran1
       PARAMETER (PI=3.141592654)
       INTEGER j,nold
       REAL am,em,en,g,oldg,p,pc,pclog,plog,pold,sq,t,y,gammln,ran1
       SAVE nold,pold,pc,plog,pclog,en,oldg
       DATA nold /-1/, pold /-1./
       if(pp.le.0.5)then
       p=pp
       else
       p=1.-pp
       endif
       am=n*p
       if (n.lt.25)then
       bnldev=0.
       do 11 j=1,n
       if(ran1(idum).lt.p)bnldev=bnldev+1.
11     continue
       else if (am.lt.1.) then
       g=exp(-am)
       t=1.
       do 12 j=0,n
       t=t*ran1(idum)
       if (t.lt.g) goto 1
12     continue
       j=n
1      bnldev=j
```

```
      else
      if (n.ne.nold) then
      en=n
      oldg=gammln(en+1.)
      nold=n
      endif
      if (p.ne.pold) then
      pc=1.-p
      plog=log(p)
      pclog=log(pc)
      pold=p
      endif
      sq=sqrt(2.*am*pc)
2     y=tan(PI*ran1(idum))
      em=sq*y+am
      if (em.lt.0..or.em.ge.en+1.) goto 2
      em=int(em)
      t=1.2*sq*(1.+y**2)*exp(oldg-gammln(em+1.)-gammln(en-em+1.)
     1   +em*plog+(en-em)*pclog)
      if (ran1(idum).gt.t) goto 2
      bnldev=em
      endif
      if (p.ne.pp) bnldev=n-bnldev
      return
      END
C  (C) Copr. 1986-92 Numerical Recipes Software 0Q-815='.
      FUNCTION gammln(xx)
c
c       Function used by Amoeba routine
c
      REAL gammln,xx
      INTEGER j
      DOUBLE PRECISION ser,stp,tmp,x,y,cof(6)
      SAVE cof,stp
      DATA cof,stp/76.18009172947146d0,-86.50532032941677d0,
     1 24.01409824083091d0,-1.231739572450155d0,.1208650973866179d-2,
     2 -.5395239384953d-5,2.5066282746310005d0/
      x=xx
      y=x
      tmp=x+5.5d0
      tmp=(x+0.5d0)*log(tmp)-tmp
      ser=1.000000000190015d0
      do 11 j=1,6
      y=y+1.d0
      ser=ser+cof(j)/y
11    continue
      gammln=tmp+log(stp*ser/x)
      return
      END
C  (C) Copr. 1986-92 Numerical Recipes Software 0Q-815='.
```

APPENDIX C

FORTRAN PROGRAM FOR MODEL M_t WITH THREE CAPTURE OCCASIONS

```
c      program Mt simulation
c
c      program to estimate population parameters based on actual data.
c      number of capture occasions and marks/unmarks are input for each
c      simulation estimates standard deviations of parameters. This
c       version is for model M0 - all capture probabilities are the
c       same each occasion for all animals. Two parameters to estimate:
c       N and p.
c
c
       character*10 outfile
       integer*4 n, cmarks, unmarks, oldpop, totcap, nsubj(25), cm(25),
     1   u(25), bigmdot, sumnb, nsqb, Nmin, Nmax, Nmint, Nmaxt, nsqt
c
c      np = number of parameters to estimate
c      mp = number of parameters + 1
c      matrix p holds initial values and is used in the estimating
c      routine
c
       parameter (ndim=2,mp=3,np=2)
       real*4 xguess(2), x(2), Nave, Nsq, newp(2), maxlik,
     1  xscale(2), p(mp,np), y(mp), Nlo, Nhi, sump, sumn, snp, likeli,
     2  phat, pcap, phatt(25),naveb, phatb, Navet, pmint(25), pmaxt(25),
     3 sumpt(25), psqt(25), sdpt(25), plowt(25), phit(25), pavet(25),
     4  plow, phi, pcapt(10)
c
c      Accept parameters for the simulation
c
       print 4
4      format (' Enter the Total Population')
       read(*,5) in
5      format(i4)
       print 8
8      format(' Input the number of repetitions for simulation i4')
       read(*,5) nrep
       print 2
2      format (' Enter random seed - large odd i7')
        read(*,6) idum
6       format (i7)
        print 9
9       format (' Enter output file name (10 characters)')
        read(*,11) outfile
11      format (a10)
       open (unit=10,status='unknown',file=outfile)
        nocc=3
        do 860 ipcap=1,8
        do 865 ip2 = -5,5
        do 870 ip3 = -5,5
        nocc = 3
        pcapt(1) = ipcap/10.
        pcapt(2)=(ipcap + ip2)/10.
        if(pcapt(2).lt.0.1) go to 865
        if(pcapt(2).gt. 0.8) go to 865
        pcapt(3)=(ipcap + ip3)/10.
        if(pcapt(3).lt.0.1) go to 870
```

```
        if(pcapt(3).gt. 0.8) go to 870
        if(pcapt(1) .eq.pcapt(2) .and. pcapt(1) .eq.pcapt(3)) go to 870
c
c       Initialize Variables
c
        nave = 0
        Nsq = 0
        pave = 0.
        naveb = 0
        caveb = 0.
        navet = 0
        cmaxb = 0.
        cminb = 1.
        chi = 0.
        sumpb = 0.
        paveb = 0.
        psqb = 0.
        pminb = 1.
        pmaxb = 0.
        csqb = 0.
        nsqb = 0.
        naveb = 0.
        sumnb = 0.
        nminb = 8000.
        nmaxb = 0.
        Nmin = 8000
        Nmax = 0
        Pmin = 1.
        Pmax = 0.
        Nmint = 8000
        Nmaxt = 0
        do 145 i = 1,25
        sumpt(i) = 0.
        pavet(i) = 0.
        psqt(i) = 0.
        pmint(i) = 1.
145      pmaxt(i) = 0.
        sumnt = 0
        Navet = 0.
        nsqt = 0
c
c      simulate mark/recapture
c
       do 50 ijk = 1,nrep
       do 100, j=1,nocc
       if (j .eq. 1) unmarks = in
c      Capture Portion
       if (j .eq. 1) then
       mdot = 0
       bigmdot = 0
        acm = bnldev(pcapt(1), unmarks, idum)
       cmarks = nint(acm)
       totcap = cmarks
       cm(1) = 0
       u(1)  = cmarks
       unmarks = unmarks - cmarks
       else
       bigmdot = bigmdot + cmarks
       acm = bnldev(pcapt(j),cmarks,idum)
       cm(j) = nint(acm)
       mdot = mdot + cm(j)
       acu = bnldev(pcapt(j),unmarks,idum)
       u(j)  = nint(acu)
```

```
      cmarks = cmarks + u(j)
      unmarks = unmarks - u(j)
      totcap = totcap + cm(j) + u(j)
      endif
100   continue
c       print *, totcap, cmarks
c     print 901, (u(i),i=1,10)
c     print 901, (cm(i), i=1,10)
901   format (1x,10(1x,i4.0))
c
c     End of simulation portion.  Estimate parameters for model M0.
c
       nlim = 2*in
       maxlik = 0
       do 200 nhat = cmarks, nlim
       likeli = alnfact(nhat,cmarks) +
     1      totcap*log(real(totcap)) +
     2      (nocc*nhat - totcap)*log(real(nocc*nhat - totcap)) -
     3      nocc*nhat*log(real(nocc*nhat))
       if (likeli .ge. maxlik) then
          maxlik = likeli
          go to 200
       else
          go to 250
       endif
200    continue
250    continue
      nhat = nhat - 1
      phat = real(totcap)/real(nocc*nhat)
c
c         Now, compute estimates for model Mt
c
       nlim = 2*in
       maxlik = 0
       do 210 nhatt = cmarks, nlim
       sum1 = 0
       sum2 = 0
       do 220 j=1, nocc
       anj = real(u(j) + cm(j))
       sum1 = sum1 + anj*log(anj)
       sum2 = sum2 + (nhatt - anj)*log(real(nhatt)-anj)
220    continue
       likeli = alnfact(nhatt,cmarks) + sum1 + sum2 -
     1      nocc*nhatt*log(real(nhatt))
       if (likeli .ge. maxlik) then
          maxlik = likeli
          go to 210
       else
          go to 240
       endif
210    continue
240    continue
      nhatt = nhatt - 1
      do 222  j=1, nocc
222   phatt(j) = (u(j)+cm(j))/real(nhatt)

c
c         Now, compute estimates for model Mb
c
       nlim = 2*in
       maxlik = 0
       do 235 nhatb = cmarks, nlim
       likeli = alnfact(nhatb,cmarks) + cmarks*log(real(cmarks)) +
```

```
     1  (nocc*nhatb-bigmdot-cmarks)*log(real(nocc*nhatb-bigmdot-cmarks))
     2   -(nocc*nhatb-bigmdot)*log(real(nocc*nhatb-bigmdot))
        if (likeli .ge. maxlik) then
           maxlik = likeli
           go to 235
        else
           go to 245
        endif
235     continue
245     continue
       nhatb = nhatb - 1
       phatb = real(cmarks)/real(nocc*nhatb-bigmdot)
       chat = real(mdot)/real(bigmdot)
c
c      print estimates for M0
c
35     format(' M0 N hat = ', i5, ' phat = ', f6.3)
c            write (10,900) phat, nhat
900   format ('M0 estimates P = ',f7.5,' N = ',i5)
      if (nhat .le. Nmin) Nmin = nhat
      if (nhat .ge. Nmax) Nmax = nhat
      sumn = sumn + nhat
      nave = nave + nhat/float(nrep)
      nsq  = nsq + nhat**2
      if (phat .le. Pmin) Pmin = phat
      if (phat .ge. Pmax) Pmax = phat
      sump = sump + phat
      Pave = Pave + phat/float(nrep)
      Psq  = Psq + phat**2
      Snp = snp + phat*nhat
c
c      print estimates for model Mt
c
c        print 45, nhatt, (phatt(i), i=1,nocc)
45      format(' N hatt = ', i5, ' phatt = ', 11(f6.3, 1x))
c            write (10,45) phatt, nhatt
      if (nhatt .le. Nmint) Nmint = nhatt
      if (nhatt .ge. Nmaxt) Nmaxt = nhatt
      sumnt = sumnt + nhatt
      Navet = Navet + nhatt/float(nrep)
      Nsqt  = Nsqt + nhatt**2
        do 29 j = 1, nocc
      if (phatt(j) .le. pmint(j)) pmint(j) = phatt(j)
      if (phatt(j) .ge. pmaxt(j)) pmaxt(j) = phatt(j)
      sumpt(j) = sumpt(j) + phatt(j)
      pavet(j) = pavet(j) + phatt(j)/float(nrep)
29    psqt(j)  = psqt(j) + phatt(j)**2
c     Snpt = snpt + phatt*nhatt
c
c      print estimates for model Mb
c
c        print 55, nhatb, phatb, chat
55     format(' N hatb = ', i5, ' phatb = ',f6.3, ' chat = ',f6.3)
c            write (10,55) nhatb, phatb, chat
      if (nhatb .le. nminb) nminb = nhatb
      if (nhatb .ge. nmaxb) nmaxb = nhatb
      sumnb = sumnb + nhatb
      naveb = naveb + real(nhatb)/real(nrep)
      nsqb  = nsqb + nhatb**2
      if (phatb .le. pminb) pminb = phatb
      if (phatb .ge. pmaxb) pmaxb = phatb
      sumpb = sumpb + phatb
      paveb = paveb + phatb/float(nrep)
```

```
      psqb  = psqb + phatb**2
      Snpb = snpb + phatb*real(nhatb)
       if (chat .le. Cminb) Cminb = chat
      if (chat .ge. Cmaxb) Cmaxb = chat
      sumc = sumc + chat
      caveb = caveb + chat/float(nrep)
      csqb  = csqb + chat**2
      sncb = sncb + chat*nhatb

50     continue
             arep = float(nrep)
c
c       Wrap up for M0
c
c        write(10,503)
503     format(///, '                      Summary for Simulation', //)
      SDN = sqrt((Nsq - arep*Nave**2)/(arep - 1.0))
      SdP = sqrt((Psq - arep *Pave**2)/(arep - 1.0))
      Nlo = nave - 1.96*SDN
      if (Nlo .le. 0.0) Nlo = 0.0
      Nhi = nave + 1.96*SDN
      plow   = pave - 1.96*SdP
      if (plow .le. 0.0) plow = 0.0
      phi = pave + 1.96*SdP
      cornp = (snp - (sumn*sump)/arep)/(
     1  (nsq -sumn**2/arep)*(psq-sump**2/arep))**.5
c      print 31, nave, SdN, Nlo, Nhi
31     format( 'M0 Nhat= ',f8.3, ' sd(N) = ',f8.4,' Low 95% = ',f7.3,
     1 ' Hi 95% = ', f8.3)
c      print 33, Pave, SdP, Plow, phi
33     format( 'M0 P hat= ',f7.3, ' sd(P) = ',f8.4,' Low 95% = ',f7.3,
     1  ' Hi 95% = ', f10.3)
c      print 34, cornp
34     format ('M0 corr n,p = ',f9.4)
c      write (10,31) nave, SdN, Nlo, Nhi
c      write (10,52) Nmin, Nmax
c      write (10,33) Pave, SdP, Plow, phi
c      write (10,58) Pmin, Pmax
c
c       Do wrap up for Mt and Mb
c
c       Wrap up for Mt
c
      SdNt = ((Nsqt - arep*Navet**2)/(arep - 1.0))**.5
      Nlo = navet - 1.96*SDNt
      if (Nlo .le. 0.0) Nlo = 0.0
      Nhi = navet + 1.96*SDNt
c      print 41, nave, SdN, Nlo, Nhi
41     format( 'Mt Nhat= ',f8.3, ' sd(N) = ',f8.4,' Low 95% = ',f7.3,
     1 ' Hi 95% = ', f8.3)
c      write (10,41) navet, SdNt, Nlo, Nhi
c      write(10,52) Nmint, Nmaxt
      do 49 j = 1,nocc
       sdpt(j) = ((psqt(j) - arep *pavet(j)**2)/(arep - 1.0))**.5
      plowt(j) = pavet(j) - 1.96*sdpt(j)
      if (plowt(j) .le. 0.0) plowt(j) = 0.0
      phit(j) = pavet(j) + 1.96*sdpt(j)
c      cornp = (snp - (sumn*sump)/arep)/(
c    1  (nsq -sumn**2/arep)*(psq-sump**2/arep))**.5
c      print 43, j, pavet(j), sdpt(j), plowt(j), phit(j)
43     format( 'Mt P hat(',i2,')= ',f9.4, ' sd(P) = ',f9.4,
     1  ' Low 95% = ',f9.4,' Hi 95% = ', f9.4)
c       print 58, pmint(j), pmaxt(j)
```

```
c      print 44, cornp
44     format ('Mt corr n,p = ',f9.4)
c      write (10,43) j, pavet(j), sdpt(j), plowt(j), phit(j)
c      write (10,58) pmint(j), pmaxt(j)
49      continue
c
c       Wrap up for Mb
c
       SDNb = ((nsqb - arep*naveb**2)/(arep - 1.0))**.5
       SdP = ((psqb - arep*paveb**2)/(arep - 1.0))**.5
       SdC = ((csqb - arep*caveb**2)/(arep - 1.0))**.5
       Nlo = naveb - 1.96*SDN
       if (Nlo .le. 0.0) Nlo = 0.0
       Nhi = naveb + 1.96*SDN
       plow = paveb - 1.96*SdP
       if (plow .le. 0.0) plow = 0.0
       phi = paveb + 1.96*SdP
       cornp = (snpb - (sumnb*sumpb)/arep)/(
     1  (nsqb -sumnb**2/arep)*(psqb-sumpb**2/arep))**.5
        clow   = caveb - 1.96*SdC
       if (clow .le. 0.0) clow = 0.0
       chi = caveb + 1.96*SdC
       cornc = (sncb - (sumnb*sumc)/arep)/(
     1  (nsqb -sumnb**2/arep)*(csqb-sumc**2/arep))**.5
c      print 51, naveb, SdN, Nlo, Nhi
51     format( 'Mb N hat= ',f8.3, ' sd(N) = ',f8.4,' Low 95% = ',f7.3,
     1 ' Hi 95% = ', f8.3)
c      print 53, Paveb, SdP, Plow, phi
53     format( 'Mb P hat= ',f9.4, ' sd(P) = ',f8.4,' Low 95% = ',f7.3,
     1  ' Hi 95% = ', f10.3)
c        print 56, caveb, SdC, clow, chi
56     format( 'Mb C hat= ',f9.4, ' sd(C) = ',f8.4,' Low 95% = ',f7.3,
     1  ' Hi 95% = ', f10.3)
c      write (10,51) naveb, SdN, Nlo, Nhi
c      write (10,52) nminb, nmaxb
52      format (' Min = ', i5, ' Max = ', i5)
c      write (10,53) Paveb, SdP, plow, phi
c      write (10,58) pminb, pmaxb
58      format(' Min = ', f9.5, ' Max = ', f9.5)
c      write (10,56) caveb, SdC, clow, chi
c      write (10, 58) cminb, cmaxb
c      write (10,57) cornp, cornc
57      format( 'Mb corr (n,p) = ', f8.5, ' corr(n,c) = ', f8.5)
c
        write(10,840) nocc, (pcapt(k), k=1,nocc), nave, SDN, pave,
     1    naveb, SDNb, paveb, caveb, navet, SdNt, (pavet(j), j=1,nocc)
840     format(i2, 18(1x,f8.3))
870     continue
865     continue
860    continue
850    continue
        print 75
75      format (' Finished!')
       close(unit=10,disp='keep')
       end

        Function alnfact(nhat,cmarks)
c
c       Function to compute ln((Nhat)!/(Nhat-cmarks)!)
c
        integer cmarks
        afact=0.0
        bfact = 0.0
```

```
        do 1 j = 1,nhat
        afact = afact + log(real(j))
1       continue
        ik = nhat - cmarks
        Do 2 j = 1,ik
2       bfact = bfact + log(real(ik))
        alnfact = afact - bfact
        return
        end
c
       FUNCTION ran1(idum)
c
c       Function used by Binomal random variate function
c
       INTEGER idum,IA,IM,IQ,IR,NTAB,NDIV
       REAL ran1,AM,EPS,RNMX
       PARAMETER (IA=16807,IM=2147483647,AM=1./IM,IQ=127773,IR=2836,
     1  NTAB=32,NDIV=1+(IM-1)/NTAB,EPS=1.2e-7,RNMX=1.-EPS)
       INTEGER j,k,iv(NTAB),iy
       SAVE iv,iy
       DATA iv /NTAB*0/, iy /0/
c      if (idum.le.0.or.iy.eq.0) then
c      idum=max(-idum,1)
       do 11 j=NTAB+8,1,-1
       k=idum/IQ
       idum=IA*(idum-k*IQ)-IR*k
       if (idum.lt.0) idum=idum+IM
       if (j.le.NTAB) iv(j)=idum
11     continue
       iy=iv(1)
c      endif
       k=idum/IQ
       idum=IA*(idum-k*IQ)-IR*k
       if (idum.lt.0) idum=idum+IM
       j=1+iy/NDIV
       iy=iv(j)
       iv(j)=idum
       ran1=min(AM*iy,RNMX)
       return
       END
C        (C) Copr. 1986-92 Numerical Recipes Software 0Q-815='.

       FUNCTION bnldev(pp,n,idum)
C       Subroutine to generate binomial rancom observations for capture
C         simulation
C
       INTEGER idum,n
       REAL*4 bnldev,pp,PI
C    USES gammln,ran1
       PARAMETER (PI=3.141592654)
       INTEGER j,nold
       REAL am,em,en,g,oldg,p,pc,pclog,plog,pold,sq,t,y,gammln,ran1
       SAVE nold,pold,pc,plog,pclog,en,oldg
       DATA nold /-1/, pold /-1./
       if(pp.le.0.5)then
       p=pp
       else
       p=1.-pp
       endif
       am=n*p
       if (n.lt.25)then
       bnldev=0.
       do 11 j=1,n
```

```
      if(ran1(idum).lt.p)bnldev=bnldev+1.
11    continue
      else if (am.lt.1.) then
      g=exp(-am)
      t=1.
      do 12 j=0,n
      t=t*ran1(idum)
      if (t.lt.g) goto 1
12    continue
      j=n
1     bnldev=j
      else
      if (n.ne.nold) then
      en=n
      oldg=gammln(en+1.)
      nold=n
      endif
      if (p.ne.pold) then
      pc=1.-p
      plog=log(p)
      pclog=log(pc)
      pold=p
      endif
      sq=sqrt(2.*am*pc)
2     y=tan(PI*ran1(idum))
      em=sq*y+am
      if (em.lt.0..or.em.ge.en+1.) goto 2
      em=int(em)
      t=1.2*sq*(1.+y**2)*exp(oldg-gammln(em+1.)-gammln(en-em+1.)
     1    +em*plog+(en-em)*pclog)
      if (ran1(idum).gt.t) goto 2
      bnldev=em
      endif
      if (p.ne.pp) bnldev=n-bnldev
      return
      END
C  (C) Copr. 1986-92 Numerical Recipes Software 0Q-815='.
      FUNCTION gammln(xx)
c
c       Function used by Amoeba routine
c
      REAL gammln,xx
      INTEGER j
      DOUBLE PRECISION ser,stp,tmp,x,y,cof(6)
      SAVE cof,stp
      DATA cof,stp/76.18009172947146d0,-86.50532032941677d0,
     1  24.01409824083091d0,-1.231739572450155d0,.1208650973866179d-2,
     2  -.5395239384953d-5,2.5066282746310005d0/
      x=xx
      y=x
      tmp=x+5.5d0
      tmp=(x+0.5d0)*log(tmp)-tmp
      ser=1.000000000190015d0
      do 11 j=1,6
      y=y+1.d0
      ser=ser+cof(j)/y
11    continue
      gammln=tmp+log(stp*ser/x)
      return
      END
C  (C) Copr. 1986-92 Numerical Recipes Software 0Q-815='.
```

APPENDIX D

FORTRAN PROGRAM FOR MODEL M_t WITH FOUR CAPTURE OCCASIONS

```
c      program Mt simulation
c
c      program to estimate population parameters based on actual data.
c      number of capture occasions and marks/unmarks are input for each
c      simulation estimates standard deviations of parameters. This
c       version is for model M0 - all capture probabilities are the
c       same each occasion for all animals. Two parameters to estimate:
c       N and p.
c
c
       character*10 outfile
       integer*4 n, cmarks, unmarks, oldpop, totcap, nsubj(25), cm(25),
     1   u(25), bigmdot, sumnb, nsqb, Nmin, Nmax, Nmint, Nmaxt, nsqt
c
c      np = number of parameters to estimate
c      mp = number of parameters + 1
c      matrix p holds initial values and is used in the estimating
c      routine
c
       parameter (ndim=2,mp=3,np=2)
       real*4 xguess(2), x(2), Nave, Nsq, newp(2), maxlik,
     1  xscale(2), p(mp,np), y(mp), Nlo, Nhi, sump, sumn, snp, likeli,
     2  phat, pcap, phatt(25),naveb, phatb, Navet, pmint(25), pmaxt(25),
     3 sumpt(25), psqt(25), sdpt(25), plowt(25), phit(25), pavet(25),
     4  plow, phi, pcapt(10)
c
c      Accept parameters for the simulation
c
       print 4
4      format (' Enter the Total Population')
       read(*,5) in
5      format(i4)
       print 8
8      format(' Input the number of repetitions for simulation i4')
       read(*,5) nrep
       print 2
2      format (' Enter random seed - large odd i7')
        read(*,6) idum
6       format (i7)
        print 9
9       format (' Enter output file name (10 characters)')
        read(*,11) outfile
11      format (a10)
       open (unit=10,status='unknown',file=outfile)
        nocc=4
        do 860 ipcap=1,8
        do 865 ip2 = -5,5
        do 870 ip3 = -5,5
        do 875 ip4 = -5,5
        print 999, ipcap
999     format (' pcap1 = ', i2)
        nocc = 4
        pcapt(1) = ipcap/10.
        pcapt(2)=(ipcap + ip2)/10.
        if(pcapt(2).lt.0.1) go to 865
        if(pcapt(2).gt. 0.8) go to 865
```

```
         pcapt(3)=(ipcap + ip3)/10.
         if(pcapt(3).lt.0.1) go to 870
         if(pcapt(3).gt. 0.8) go to 870
         pcapt(4)=(ipcap + ip4)/10.
         if(pcapt(4).lt.0.1) go to 875
         if(pcapt(4).gt. 0.8) go to 875
         if(pcapt(1).eq.pcapt(2) .and. pcapt(1).eq.pcapt(3) .and.
     1    pcapt(1).eq.pcapt(4)) go to 875
c
c        Initialize Variables
c
         nave = 0
         Nsq = 0
         pave = 0.
         naveb = 0
         caveb = 0.
         navet = 0
         cmaxb = 0.
         cminb = 1.
         chi = 0.
         sumpb = 0.
         paveb = 0.
         psqb = 0.
         pminb = 1.
         pmaxb = 0.
         csqb = 0.
         nsqb = 0.
         naveb = 0.
         sumnb = 0.
         nminb = 8000.
         nmaxb = 0.
         Nmin = 8000
         Nmax = 0
         Pmin = 1.
         Pmax = 0.
         Nmint = 8000
         Nmaxt = 0
         do 145 i = 1,25
         sumpt(i) = 0.
         pavet(i) = 0.
         psqt(i) = 0.
         pmint(i) = 1.
145       pmaxt(i) = 0.
         sumnt = 0
         Navet = 0.
         nsqt = 0
c
c       simulate mark/recapture
c
        do 50 ijk = 1,nrep
        do 100, j=1,nocc
        if (j .eq. 1) unmarks = in
c       Capture Portion
        if (j .eq. 1) then
        mdot = 0
        bigmdot = 0
         acm = bnldev(pcapt(1), unmarks, idum)
        cmarks = nint(acm)
        totcap = cmarks
        cm(1) = 0
        u(1)  = cmarks
        unmarks = unmarks - cmarks
        else
```

```
      bigmdot = bigmdot + cmarks
      acm = bnldev(pcapt(j),cmarks,idum)
      cm(j) = nint(acm)
      mdot = mdot + cm(j)
      acu = bnldev(pcapt(j),unmarks,idum)
      u(j)  = nint(acu)
      cmarks = cmarks + u(j)
      unmarks = unmarks - u(j)
      totcap = totcap + cm(j) + u(j)
      endif
100   continue
c      print *, totcap, cmarks
c     print 901, (u(i),i=1,10)
c     print 901, (cm(i), i=1,10)
901   format (1x,10(1x,i4.0))
c
c     End of simulation portion.  Estimate parameters for model M0.
c
       nlim = 2*in
       maxlik = 0
       do 200 nhat = cmarks, nlim
       likeli = alnfact(nhat,cmarks) +
     1     totcap*log(real(totcap)) +
     2     (nocc*nhat - totcap)*log(real(nocc*nhat - totcap)) -
     3     nocc*nhat*log(real(nocc*nhat))
       if (likeli .ge. maxlik) then
          maxlik = likeli
          go to 200
       else
          go to 250
       endif
200    continue
250    continue
      nhat = nhat - 1
      phat = real(totcap)/real(nocc*nhat)
c
c         Now, compute estimates for model Mt
c
       nlim = 2*in
       maxlik = 0
       do 210 nhatt = cmarks, nlim
       sum1 = 0
       sum2 = 0
       do 220 j=1, nocc
       anj = real(u(j) + cm(j))
       sum1 = sum1 + anj*log(anj)
       sum2 = sum2 + (nhatt - anj)*log(real(nhatt)-anj)
220    continue
       likeli = alnfact(nhatt,cmarks) + sum1 + sum2 -
     1     nocc*nhatt*log(real(nhatt))
       if (likeli .ge. maxlik) then
          maxlik = likeli
          go to 210
       else
          go to 240
       endif
210    continue
240    continue
      nhatt = nhatt - 1
      do 222  j=1, nocc
222   phatt(j) = (u(j)+cm(j))/real(nhatt)

c
```

```
c          Now, compute estimates for model Mb
c
        nlim = 2*in
        maxlik = 0
        do 235 nhatb = cmarks, nlim
        likeli = alnfact(nhatb,cmarks) + cmarks*log(real(cmarks)) +
     1  (nocc*nhatb-bigmdot-cmarks)*log(real(nocc*nhatb-bigmdot-cmarks))
     2   -(nocc*nhatb-bigmdot)*log(real(nocc*nhatb-bigmdot))
        if (likeli .ge. maxlik) then
           maxlik = likeli
           go to 235
        else
           go to 245
        endif
235     continue
245     continue
       nhatb = nhatb - 1
       phatb = real(cmarks)/real(nocc*nhatb-bigmdot)
       chat = real(mdot)/real(bigmdot)
c
c      print estimates for M0
c
35     format(' M0 N hat = ', i5, ' phat = ', f6.3)
c            write (10,900) phat, nhat
900    format ('M0 estimates P = ',f7.5,' N = ',i5)
       if (nhat .le. Nmin) Nmin = nhat
       if (nhat .ge. Nmax) Nmax = nhat
       sumn = sumn + nhat
       nave = nave + nhat/float(nrep)
       nsq  = nsq + nhat**2
       if (phat .le. Pmin) Pmin = phat
       if (phat .ge. Pmax) Pmax = phat
       sump = sump + phat
       Pave = Pave + phat/float(nrep)
       Psq  = Psq + phat**2
       Snp = snp + phat*nhat
c
c      print estimates for model Mt
c
c        print 45, nhatt, (phatt(i), i=1,nocc)
45      format(' N hatt = ', i5, ' phatt = ', 11(f6.3, 1x))
c            write (10,45) phatt, nhatt
       if (nhatt .le. Nmint) Nmint = nhatt
       if (nhatt .ge. Nmaxt) Nmaxt = nhatt
       sumnt = sumnt + nhatt
       Navet = Navet + nhatt/float(nrep)
       Nsqt  = Nsqt + nhatt**2
        do 29 j = 1, nocc
       if (phatt(j) .le. pmint(j)) pmint(j) = phatt(j)
       if (phatt(j) .ge. pmaxt(j)) pmaxt(j) = phatt(j)
       sumpt(j) = sumpt(j) + phatt(j)
       pavet(j) = pavet(j) + phatt(j)/float(nrep)
29     psqt(j)  = psqt(j) + phatt(j)**2
c      Snpt = snpt + phatt*nhatt
c
c      print estimates for model Mb
c
c        print 55, nhatb, phatb, chat
55     format(' N hatb = ', i5, ' phatb = ',f6.3, ' chat = ',f6.3)
c            write (10,55) nhatb, phatb, chat
       if (nhatb .le. nminb) nminb = nhatb
       if (nhatb .ge. nmaxb) nmaxb = nhatb
       sumnb = sumnb + nhatb
```

```
      naveb = naveb + real(nhatb)/real(nrep)
      nsqb  = nsqb + nhatb**2
      if (phatb .le. pminb) pminb = phatb
      if (phatb .ge. pmaxb) pmaxb = phatb
      sumpb = sumpb + phatb
      paveb = paveb + phatb/float(nrep)
      psqb  = psqb + phatb**2
      Snpb = snpb + phatb*real(nhatb)
       if (chat .le. Cminb) Cminb = chat
      if (chat .ge. Cmaxb) Cmaxb = chat
      sumc = sumc + chat
      caveb = caveb + chat/float(nrep)
      csqb  = csqb + chat**2
      sncb = sncb + chat*nhatb

50     continue
             arep = float(nrep)
c
c       Wrap up for M0
c
c        write(10,503)
503     format(///, '                     Summary for Simulation', //)
      SDN = sqrt((Nsq - arep*Nave**2)/(arep - 1.0))
      SdP = sqrt((Psq - arep *Pave**2)/(arep - 1.0))
      Nlo = nave - 1.96*SDN
      if (Nlo .le. 0.0) Nlo = 0.0
      Nhi = nave + 1.96*SDN
      plow   = pave - 1.96*SdP
      if (plow .le. 0.0) plow = 0.0
      phi = pave + 1.96*SdP
      cornp = (snp - (sumn*sump)/arep)/(
     1  (nsq -sumn**2/arep)*(psq-sump**2/arep))**.5
c     print 31, nave, SdN, Nlo, Nhi
31    format( 'M0 Nhat= ',f8.3, ' sd(N) = ',f8.4,' Low 95% = ',f7.3,
     1 ' Hi 95% = ', f8.3)
c     print 33, Pave, SdP, Plow, phi
33    format( 'M0 P hat= ',f7.3, ' sd(P) = ',f8.4,' Low 95% = ',f7.3,
     1  ' Hi 95% = ', f10.3)
c     print 34, cornp
34    format ('M0 corr n,p = ',f9.4)
c     write (10,31) nave, SdN, Nlo, Nhi
c     write (10,52) Nmin, Nmax
c     write (10,33) Pave, SdP, Plow, phi
c     write (10,58) Pmin, Pmax
c
c       Do wrap up for Mt and Mb
c
c       Wrap up for Mt
c
      SdNt = ((Nsqt - arep*Navet**2)/(arep - 1.0))**.5
      Nlo = navet - 1.96*SDNt
      if (Nlo .le. 0.0) Nlo = 0.0
      Nhi = navet + 1.96*SDNt
c     print 41, nave, SdN, Nlo, Nhi
41    format( 'Mt Nhat= ',f8.3, ' sd(N) = ',f8.4,' Low 95% = ',f7.3,
     1 ' Hi 95% = ', f8.3)
c     write (10,41) navet, SdNt, Nlo, Nhi
c     write(10,52) Nmint, Nmaxt
      do 49 j = 1,nocc
       sdpt(j) = ((psqt(j) - arep *pavet(j)**2)/(arep - 1.0))**.5
      plowt(j) = pavet(j) - 1.96*sdpt(j)
      if (plowt(j) .le. 0.0) plowt(j) = 0.0
      phit(j) = pavet(j) + 1.96*sdpt(j)
```

```
c      cornp = (snp - (sumn*sump)/arep)/(
c    1  (nsq -sumn**2/arep)*(psq-sump**2/arep))**.5
c      print 43, j, pavet(j), sdpt(j), plowt(j), phit(j)
43     format( 'Mt P hat(',i2,')= ',f9.4, ' sd(P) = ',f9.4,
     1  ' Low 95% = ',f9.4,' Hi 95% = ', f9.4)
c       print 58, pmint(j), pmaxt(j)
c      print 44, cornp
44     format ('Mt corr n,p = ',f9.4)
c      write (10,43) j, pavet(j), sdpt(j), plowt(j), phit(j)
c      write (10,58) pmint(j), pmaxt(j)
49      continue
c
c       Wrap up for Mb
c
       SDNb = ((nsqb - arep*naveb**2)/(arep - 1.0))**.5
       SdP = ((psqb - arep*paveb**2)/(arep - 1.0))**.5
       SdC = ((csqb - arep*caveb**2)/(arep - 1.0))**.5
       Nlo = naveb - 1.96*SDN
       if (Nlo .le. 0.0) Nlo = 0.0
       Nhi = naveb + 1.96*SDN
       plow = paveb - 1.96*SdP
       if (plow .le. 0.0) plow = 0.0
       phi = paveb + 1.96*SdP
       cornp = (snpb - (sumnb*sumpb)/arep)/(
     1  (nsqb -sumnb**2/arep)*(psqb-sumpb**2/arep))**.5
        clow   = caveb - 1.96*SdC
       if (clow .le. 0.0) clow = 0.0
       chi = caveb + 1.96*SdC
       cornc = (sncb - (sumnb*sumc)/arep)/(
     1  (nsqb -sumnb**2/arep)*(csqb-sumc**2/arep))**.5
c      print 51, naveb, SdN, Nlo, Nhi
51     format( 'Mb N hat= ',f8.3, ' sd(N) = ',f8.4,' Low 95% = ',f7.3,
     1 ' Hi 95% = ', f8.3)
c      print 53, Paveb, SdP, Plow, phi
53     format( 'Mb P hat= ',f9.4, ' sd(P) = ',f8.4,' Low 95% = ',f7.3,
     1  ' Hi 95% = ', f10.3)
c        print 56, caveb, SdC, clow, chi
56     format( 'Mb C hat= ',f9.4, ' sd(C) = ',f8.4,' Low 95% = ',f7.3,
     1  ' Hi 95% = ', f10.3)
c      write (10,51) naveb, SdN, Nlo, Nhi
c      write (10,52) nminb, nmaxb
52      format (' Min = ', i5, ' Max = ', i5)
c      write (10,53) Paveb, SdP, plow, phi
c      write (10,58) pminb, pmaxb
58      format(' Min = ', f9.5, ' Max = ', f9.5)
c      write (10,56) caveb, SdC, clow, chi
c      write (10, 58) cminb, cmaxb
c      write (10,57) cornp, cornc
57      format( 'Mb corr (n,p) = ', f8.5, ' corr(n,c) = ', f8.5)
c
        write(10,840) nocc, (pcapt(k), k=1,nocc), nave, SDN, pave,
     1    naveb, SDNb, paveb, caveb, navet, SdNt, (pavet(j), j=1,nocc)
840     format(i2, 18(1x,f8.3))
875     continue
870     continue
865     continue
860    continue
850    continue
        print 75
75      format (' Finished!')
       close(unit=10,disp='keep')
        end
```

```
        Function alnfact(nhat,cmarks)
c
c       Function to compute ln((Nhat)!/(Nhat-cmarks)!)
c
        integer cmarks
        afact=0.0
        bfact = 0.0
        do 1 j = 1,nhat
        afact = afact + log(real(j))
1       continue
        ik = nhat - cmarks
        Do 2 j = 1,ik
2       bfact = bfact + log(real(ik))
        alnfact = afact - bfact
        return
        end
c
      FUNCTION ran1(idum)
c
c       Function used by Binomal random variate function
c
      INTEGER idum,IA,IM,IQ,IR,NTAB,NDIV
      REAL ran1,AM,EPS,RNMX
      PARAMETER (IA=16807,IM=2147483647,AM=1./IM,IQ=127773,IR=2836,
     1  NTAB=32,NDIV=1+(IM-1)/NTAB,EPS=1.2e-7,RNMX=1.-EPS)
      INTEGER j,k,iv(NTAB),iy
      SAVE iv,iy
      DATA iv /NTAB*0/, iy /0/
c     if (idum.le.0.or.iy.eq.0) then
c     idum=max(-idum,1)
      do 11 j=NTAB+8,1,-1
      k=idum/IQ
      idum=IA*(idum-k*IQ)-IR*k
      if (idum.lt.0) idum=idum+IM
      if (j.le.NTAB) iv(j)=idum
11    continue
      iy=iv(1)
c     endif
      k=idum/IQ
      idum=IA*(idum-k*IQ)-IR*k
      if (idum.lt.0) idum=idum+IM
      j=1+iy/NDIV
      iy=iv(j)
      iv(j)=idum
      ran1=min(AM*iy,RNMX)
      return
      END
C        (C) Copr. 1986-92 Numerical Recipes Software 0Q-815='.

      FUNCTION bnldev(pp,n,idum)
C       Subroutine to generate binomial rancom observations for capture
C         simulation
C
      INTEGER idum,n
      REAL*4 bnldev,pp,PI
C    USES gammln,ran1
      PARAMETER (PI=3.141592654)
      INTEGER j,nold
      REAL am,em,en,g,oldg,p,pc,pclog,plog,pold,sq,t,y,gammln,ran1
      SAVE nold,pold,pc,plog,pclog,en,oldg
      DATA nold /-1/, pold /-1./
      if(pp.le.0.5)then
      p=pp
```

```
      else
      p=1.-pp
      endif
      am=n*p
      if (n.lt.25)then
      bnldev=0.
      do 11 j=1,n
      if(ran1(idum).lt.p)bnldev=bnldev+1.
11    continue
      else if (am.lt.1.) then
      g=exp(-am)
      t=1.
      do 12 j=0,n
      t=t*ran1(idum)
      if (t.lt.g) goto 1
12    continue
      j=n
1     bnldev=j
      else
      if (n.ne.nold) then
      en=n
      oldg=gammln(en+1.)
      nold=n
      endif
      if (p.ne.pold) then
      pc=1.-p
      plog=log(p)
      pclog=log(pc)
      pold=p
      endif
      sq=sqrt(2.*am*pc)
2     y=tan(PI*ran1(idum))
      em=sq*y+am
      if (em.lt.0..or.em.ge.en+1.) goto 2
      em=int(em)
      t=1.2*sq*(1.+y**2)*exp(oldg-gammln(em+1.)-gammln(en-em+1.)
     1    +em*plog+(en-em)*pclog)
      if (ran1(idum).gt.t) goto 2
      bnldev=em
      endif
      if (p.ne.pp) bnldev=n-bnldev
      return
      END
C  (C) Copr. 1986-92 Numerical Recipes Software 0Q-815='.
      FUNCTION gammln(xx)
c
c       Function used by Amoeba routine
c
      REAL gammln,xx
      INTEGER j
      DOUBLE PRECISION ser,stp,tmp,x,y,cof(6)
      SAVE cof,stp
      DATA cof,stp/76.18009172947146d0,-86.50532032941677d0,
     1  24.01409824083091d0,-1.231739572450155d0,.1208650973866179d-2,
     2  -.5395239384953d-5,2.5066282746310005d0/
      x=xx
      y=x
      tmp=x+5.5d0
      tmp=(x+0.5d0)*log(tmp)-tmp
      ser=1.000000000190015d0
      do 11 j=1,6
      y=y+1.d0
      ser=ser+cof(j)/y
```

```
11     continue
       gammln=tmp+log(stp*ser/x)
       return
       END
C  (C) Copr. 1986-92 Numerical Recipes Software 0Q-815='.
```

APPENDIX E

FORTRAN PROGRAM FOR MODEL M_t WITH FIVE CAPTURE OCCASIONS

```
c      program Mt simulation
c
c      program to estimate population parameters based on actual data.
c      number of capture occasions and marks/unmarks are input for each
c      simulation estimates standard deviations of parameters. This
c        version is for model M0 - all capture probabilities are the
c        same each occasion for all animals. Two parameters to estimate:
c        N and p.
c
c
       character*10 outfile
       integer*4 n, cmarks, unmarks, oldpop, totcap, nsubj(25), cm(25),
     1   u(25), bigmdot, sumnb, nsqb, Nmin, Nmax, Nmint, Nmaxt, nsqt
c
c       np = number of parameters to estimate
c       mp = number of parameters + 1
c       matrix p holds initial values and is used in the estimating
c       routine
c
       parameter (ndim=2,mp=3,np=2)
       real*4 xguess(2), x(2), Nave, Nsq, newp(2), maxlik,
     1  xscale(2), p(mp,np), y(mp), Nlo, Nhi, sump, sumn, snp, likeli,
     2  phat, pcap, phatt(25),naveb, phatb, Navet, pmint(25), pmaxt(25),
     3 sumpt(25), psqt(25), sdpt(25), plowt(25), phit(25), pavet(25),
     4  plow, phi, pcapt(10)
c
c      Accept parameters for the simulation
c
       print 4
4      format (' Enter the Total Population')
       read(*,5) in
5      format(i4)
       print 8
8      format(' Input the number of repetitions for simulation i4')
       read(*,5) nrep
       print 2
2      format (' Enter random seed - large odd i7')
        read(*,6) idum
6       format (i7)
        print 9
9       format (' Enter output file name (10 characters)')
        read(*,11) outfile
11      format (a10)
       open (unit=10,status='unknown',file=outfile)
        nocc=4
        do 860 ipcap=1,8
         print 999, ipcap
999     format (' pcap1 = ', i2)
        do 865 ip2 = -5,5
        do 870 ip3 = -5,5
        do 875 ip4 = -5,5
        do 880 ip5 = -5,5
        nocc = 5
        pcapt(1) = ipcap/10.
        pcapt(2)=(ipcap + ip2)/10.
        if(pcapt(2).lt.0.1) go to 865
        if(pcapt(2).gt. 0.8) go to 865
        pcapt(3)=(ipcap + ip3)/10.
```

```
        if(pcapt(3).lt. 0.1) go to 870
        if(pcapt(3).gt. 0.8) go to 870
        pcapt(4)=(ipcap + ip4)/10.
        if(pcapt(4).lt. 0.1) go to 875
        if(pcapt(4).gt. 0.8) go to 875
        pcapt(5)=(ipcap + ip5)/10.
        if(pcapt(5).lt.0.1) go to 880
        if(pcapt(5).gt. 0.8) go to 880
        if(pcapt(1).eq.pcapt(2) .and. pcapt(1).eq.pcapt(3) .and.
     1   pcapt(1).eq.pcapt(4).and. pcapt(1).eq.pcapt(5)) go to 880
c
c       Initialize Variables
c
        nave = 0
        Nsq = 0
        pave = 0.
        naveb = 0
        caveb = 0.
        navet = 0
        cmaxb = 0.
        cminb = 1.
        chi = 0.
        sumpb = 0.
        paveb = 0.
        psqb = 0.
        pminb = 1.
        pmaxb = 0.
        csqb = 0.
        nsqb = 0.
        naveb = 0.
        sumnb = 0.
        nminb = 8000.
        nmaxb = 0.
        Nmin = 8000
        Nmax = 0
        Pmin = 1.
        Pmax = 0.
        Nmint = 8000
        Nmaxt = 0
        do 145 i = 1,25
        sumpt(i) = 0.
        pavet(i) = 0.
        psqt(i) = 0.
        pmint(i) = 1.
145      pmaxt(i) = 0.
        sumnt = 0
        Navet = 0.
        nsqt = 0
c
c      simulate mark/recapture
c
       do 50 ijk = 1,nrep
       do 100, j=1,nocc
       if (j .eq. 1) unmarks = in
c      Capture Portion
       if (j .eq. 1) then
       mdot = 0
       bigmdot = 0
        acm = bnldev(pcapt(1), unmarks, idum)
       cmarks = nint(acm)
       totcap = cmarks
       cm(1) = 0
       u(1)  = cmarks
```

```
        unmarks = unmarks - cmarks
        else
        bigmdot = bigmdot + cmarks
        acm = bnldev(pcapt(j),cmarks,idum)
        cm(j) = nint(acm)
        mdot = mdot + cm(j)
        acu = bnldev(pcapt(j),unmarks,idum)
        u(j)  = nint(acu)
        cmarks = cmarks + u(j)
        unmarks = unmarks - u(j)
        totcap = totcap + cm(j) + u(j)
        endif
100     continue
c         print *, totcap, cmarks
c       print 901, (u(i),i=1,10)
c       print 901, (cm(i), i=1,10)
901     format (1x,10(1x,i4.0))
c
c       End of simulation portion.  Estimate parameters for model M0.
c
         nlim = 2*in
         maxlik = 0
         do 200 nhat = cmarks, nlim
         likeli = alnfact(nhat,cmarks) +
     1       totcap*log(real(totcap)) +
     2       (nocc*nhat - totcap)*log(real(nocc*nhat - totcap)) -
     3       nocc*nhat*log(real(nocc*nhat))
         if (likeli .ge. maxlik) then
            maxlik = likeli
            go to 200
         else
            go to 250
         endif
200      continue
250      continue
        nhat = nhat - 1
        phat = real(totcap)/real(nocc*nhat)
c
c           Now, compute estimates for model Mt
c
         nlim = 2*in
         maxlik = 0
         do 210 nhatt = cmarks, nlim
         sum1 = 0
         sum2 = 0
         do 220 j=1, nocc
         anj = real(u(j) + cm(j))
         sum1 = sum1 + anj*log(anj)
         sum2 = sum2 + (nhatt - anj)*log(real(nhatt)-anj)
220      continue
         likeli = alnfact(nhatt,cmarks) + sum1 + sum2 -
     1       nocc*nhatt*log(real(nhatt))
         if (likeli .ge. maxlik) then
            maxlik = likeli
            go to 210
         else
            go to 240
         endif
210      continue
240      continue
        nhatt = nhatt - 1
        do 222  j=1, nocc
222     phatt(j) = (u(j)+cm(j))/real(nhatt)
```

```
c
c          Now, compute estimates for model Mb
c
        nlim = 2*in
        maxlik = 0
        do 235 nhatb = cmarks, nlim
        likeli = alnfact(nhatb,cmarks) + cmarks*log(real(cmarks)) +
     1  (nocc*nhatb-bigmdot-cmarks)*log(real(nocc*nhatb-bigmdot-cmarks))
     2   -(nocc*nhatb-bigmdot)*log(real(nocc*nhatb-bigmdot))
        if (likeli .ge. maxlik) then
           maxlik = likeli
           go to 235
        else
           go to 245
        endif
235     continue
245     continue
       nhatb = nhatb - 1
       phatb = real(cmarks)/real(nocc*nhatb-bigmdot)
       chat = real(mdot)/real(bigmdot)
c
c      print estimates for M0
c
35     format(' M0 N hat = ', i5, ' phat = ', f6.3)
c            write (10,900) phat, nhat
900    format ('M0 estimates P = ',f7.5,' N = ',i5)
       if (nhat .le. Nmin) Nmin = nhat
       if (nhat .ge. Nmax) Nmax = nhat
       sumn = sumn + nhat
       nave = nave + nhat/float(nrep)
       nsq  = nsq + nhat**2
       if (phat .le. Pmin) Pmin = phat
       if (phat .ge. Pmax) Pmax = phat
       sump = sump + phat
       Pave = Pave + phat/float(nrep)
       Psq  = Psq + phat**2
       Snp = snp + phat*nhat
c
c      print estimates for model Mt
c
c        print 45, nhatt, (phatt(i), i=1,nocc)
45      format(' N hatt = ', i5, ' phatt = ', 11(f6.3, 1x))
c            write (10,45) phatt, nhatt
       if (nhatt .le. Nmint) Nmint = nhatt
       if (nhatt .ge. Nmaxt) Nmaxt = nhatt
       sumnt = sumnt + nhatt
       Navet = Navet + nhatt/float(nrep)
       Nsqt  = Nsqt + nhatt**2
        do 29 j = 1, nocc
       if (phatt(j) .le. pmint(j)) pmint(j) = phatt(j)
       if (phatt(j) .ge. pmaxt(j)) pmaxt(j) = phatt(j)
       sumpt(j) = sumpt(j) + phatt(j)
       pavet(j) = pavet(j) + phatt(j)/float(nrep)
29     psqt(j)  = psqt(j) + phatt(j)**2
c      Snpt = snpt + phatt*nhatt
c
c      print estimates for model Mb
c
c        print 55, nhatb, phatb, chat
55     format(' N hatb = ', i5, ' phatb = ',f6.3, ' chat = ',f6.3)
c            write (10,55) nhatb, phatb, chat
       if (nhatb .le. nminb) nminb = nhatb
```

```
      if (nhatb .ge. nmaxb) nmaxb = nhatb
      sumnb = sumnb + nhatb
      naveb = naveb + real(nhatb)/real(nrep)
      nsqb  = nsqb + nhatb**2
      if (phatb .le. pminb) pminb = phatb
      if (phatb .ge. pmaxb) pmaxb = phatb
      sumpb = sumpb + phatb
      paveb = paveb + phatb/float(nrep)
      psqb  = psqb + phatb**2
      Snpb = snpb + phatb*real(nhatb)
       if (chat .le. Cminb) Cminb = chat
      if (chat .ge. Cmaxb) Cmaxb = chat
      sumc = sumc + chat
      caveb = caveb + chat/float(nrep)
      csqb  = csqb + chat**2
      sncb = sncb + chat*nhatb

50     continue
            arep = float(nrep)
c
c       Wrap up for M0
c
c        write(10,503)
503     format(///, '                    Summary for Simulation', //)
      SDN = sqrt((Nsq - arep*Nave**2)/(arep - 1.0))
      SdP = sqrt((Psq - arep *Pave**2)/(arep - 1.0))
      Nlo = nave - 1.96*SDN
      if (Nlo .le. 0.0) Nlo = 0.0
      Nhi = nave + 1.96*SDN
      plow   = pave - 1.96*SdP
      if (plow .le. 0.0) plow = 0.0
      phi = pave + 1.96*SdP
      cornp = (snp - (sumn*sump)/arep)/(
     1  (nsq -sumn**2/arep)*(psq-sump**2/arep))**.5
c      print 31, nave, SdN, Nlo, Nhi
31     format( 'M0 Nhat= ',f8.3, ' sd(N) = ',f8.4,' Low 95% = ',f7.3,
     1 ' Hi 95% = ', f8.3)
c      print 33, Pave, SdP, Plow, phi
33     format( 'M0 P hat= ',f7.3, ' sd(P) = ',f8.4,' Low 95% = ',f7.3,
     1  ' Hi 95% = ', f10.3)
c      print 34, cornp
34     format ('M0 corr n,p = ',f9.4)
c      write (10,31) nave, SdN, Nlo, Nhi
c      write (10,52) Nmin, Nmax
c      write (10,33) Pave, SdP, Plow, phi
c      write (10,58) Pmin, Pmax
c
c       Do wrap up for Mt and Mb
c
c       Wrap up for Mt
c
      SdNt = ((Nsqt - arep*Navet**2)/(arep - 1.0))**.5
      Nlo = navet - 1.96*SDNt
      if (Nlo .le. 0.0) Nlo = 0.0
      Nhi = navet + 1.96*SDNt
c      print 41, nave, SdN, Nlo, Nhi
41     format( 'Mt Nhat= ',f8.3, ' sd(N) = ',f8.4,' Low 95% = ',f7.3,
     1 ' Hi 95% = ', f8.3)
c      write (10,41) navet, SdNt, Nlo, Nhi
c      write(10,52) Nmint, Nmaxt
      do 49 j = 1,nocc
       sdpt(j) = ((psqt(j) - arep *pavet(j)**2)/(arep - 1.0))**.5
      plowt(j) = pavet(j) - 1.96*sdpt(j)
```

```
      if (plowt(j) .le. 0.0) plowt(j) = 0.0
      phit(j) = pavet(j) + 1.96*sdpt(j)
c     cornp = (snp - (sumn*sump)/arep)/(
c    1  (nsq -sumn**2/arep)*(psq-sump**2/arep))**.5
c     print 43, j, pavet(j), sdpt(j), plowt(j), phit(j)
43    format( 'Mt P hat(',i2,')= ',f9.4, ' sd(P) = ',f9.4,
     1  ' Low 95% = ',f9.4,' Hi 95% = ', f9.4)
c      print 58, pmint(j), pmaxt(j)
c     print 44, cornp
44    format ('Mt corr n,p = ',f9.4)
c     write (10,43) j, pavet(j), sdpt(j), plowt(j), phit(j)
c     write (10,58) pmint(j), pmaxt(j)
49     continue
c
c      Wrap up for Mb
c
      SDNb = ((nsqb - arep*naveb**2)/(arep - 1.0))**.5
      SdP = ((psqb - arep*paveb**2)/(arep - 1.0))**.5
      SdC = ((csqb - arep*caveb**2)/(arep - 1.0))**.5
      Nlo = naveb - 1.96*SDN
      if (Nlo .le. 0.0) Nlo = 0.0
      Nhi = naveb + 1.96*SDN
      plow = paveb - 1.96*SdP
      if (plow .le. 0.0) plow = 0.0
      phi = paveb + 1.96*SdP
      cornp = (snpb - (sumnb*sumpb)/arep)/(
     1  (nsqb -sumnb**2/arep)*(psqb-sumpb**2/arep))**.5
       clow   = caveb - 1.96*SdC
      if (clow .le. 0.0) clow = 0.0
      chi = caveb + 1.96*SdC
      cornc = (sncb - (sumnb*sumc)/arep)/(
     1  (nsqb -sumnb**2/arep)*(csqb-sumc**2/arep))**.5
c     print 51, naveb, SdN, Nlo, Nhi
51    format( 'Mb N hat= ',f8.3, ' sd(N) = ',f8.4,' Low 95% = ',f7.3,
     1 ' Hi 95% = ', f8.3)
c     print 53, Paveb, SdP, Plow, phi
53    format( 'Mb P hat= ',f9.4, ' sd(P) = ',f8.4,' Low 95% = ',f7.3,
     1  ' Hi 95% = ', f10.3)
c       print 56, caveb, SdC, clow, chi
56    format( 'Mb C hat= ',f9.4, ' sd(C) = ',f8.4,' Low 95% = ',f7.3,
     1  ' Hi 95% = ', f10.3)
c     write (10,51) naveb, SdN, Nlo, Nhi
c     write (10,52) nminb, nmaxb
52     format (' Min = ', i5, ' Max = ', i5)
c     write (10,53) Paveb, SdP, plow, phi
c     write (10,58) pminb, pmaxb
58     format(' Min = ', f9.5, ' Max = ', f9.5)
c     write (10,56) caveb, SdC, clow, chi
c     write (10, 58) cminb, cmaxb
c     write (10,57) cornp, cornc
57     format( 'Mb corr (n,p) = ', f8.5, ' corr(n,c) = ', f8.5)
c
       write(10,840) nocc, (pcapt(k), k=1,nocc), nave, SDN, pave,
     1    naveb, SDNb, paveb, caveb, navet, SdNt, (pavet(j), j=1,nocc)
840    format(i2, 19(1x,f8.3))
880    continue
875    continue
870    continue
865    continue
860   continue
850   continue
       print 75
75     format (' Finished!')
```

```
      close(unit=10,disp='keep')
      end

        Function alnfact(nhat,cmarks)
c
c       Function to compute ln((Nhat)!/(Nhat-cmarks)!)
c
        integer cmarks
        afact=0.0
        bfact = 0.0
        do 1 j = 1,nhat
        afact = afact + log(real(j))
1       continue
        ik = nhat - cmarks
        Do 2 j = 1,ik
2       bfact = bfact + log(real(ik))
        alnfact = afact - bfact
        return
        end
c
      FUNCTION ran1(idum)
c
c       Function used by Binomal random variate function
c
      INTEGER idum,IA,IM,IQ,IR,NTAB,NDIV
      REAL ran1,AM,EPS,RNMX
      PARAMETER (IA=16807,IM=2147483647,AM=1./IM,IQ=127773,IR=2836,
     1  NTAB=32,NDIV=1+(IM-1)/NTAB,EPS=1.2e-7,RNMX=1.-EPS)
      INTEGER j,k,iv(NTAB),iy
      SAVE iv,iy
      DATA iv /NTAB*0/, iy /0/
c     if (idum.le.0.or.iy.eq.0) then
c     idum=max(-idum,1)
      do 11 j=NTAB+8,1,-1
      k=idum/IQ
      idum=IA*(idum-k*IQ)-IR*k
      if (idum.lt.0) idum=idum+IM
      if (j.le.NTAB) iv(j)=idum
11    continue
      iy=iv(1)
c     endif
      k=idum/IQ
      idum=IA*(idum-k*IQ)-IR*k
      if (idum.lt.0) idum=idum+IM
      j=1+iy/NDIV
      iy=iv(j)
      iv(j)=idum
      ran1=min(AM*iy,RNMX)
      return
      END
C        (C) Copr. 1986-92 Numerical Recipes Software 0Q-815='.

      FUNCTION bnldev(pp,n,idum)
C       Subroutine to generate binomial rancom observations for capture
C         simulation
C
      INTEGER idum,n
      REAL*4 bnldev,pp,PI
C    USES gammln,ran1
      PARAMETER (PI=3.141592654)
      INTEGER j,nold
      REAL am,em,en,g,oldg,p,pc,pclog,plog,pold,sq,t,y,gammln,ran1
      SAVE nold,pold,pc,plog,pclog,en,oldg
```

```
      DATA nold /-1/, pold /-1./
      if(pp.le.0.5)then
      p=pp
      else
      p=1.-pp
      endif
      am=n*p
      if (n.lt.25)then
      bnldev=0.
      do 11 j=1,n
      if(ran1(idum).lt.p)bnldev=bnldev+1.
11    continue
      else if (am.lt.1.) then
      g=exp(-am)
      t=1.
      do 12 j=0,n
      t=t*ran1(idum)
      if (t.lt.g) goto 1
12    continue
      j=n
1     bnldev=j
      else
      if (n.ne.nold) then
      en=n
      oldg=gammln(en+1.)
      nold=n
      endif
      if (p.ne.pold) then
      pc=1.-p
      plog=log(p)
      pclog=log(pc)
      pold=p
      endif
      sq=sqrt(2.*am*pc)
2     y=tan(PI*ran1(idum))
      em=sq*y+am
      if (em.lt.0..or.em.ge.en+1.) goto 2
      em=int(em)
      t=1.2*sq*(1.+y**2)*exp(oldg-gammln(em+1.)-gammln(en-em+1.)
     1    +em*plog+(en-em)*pclog)
      if (ran1(idum).gt.t) goto 2
      bnldev=em
      endif
      if (p.ne.pp) bnldev=n-bnldev
      return
      END
C  (C) Copr. 1986-92 Numerical Recipes Software 0Q-815='.
      FUNCTION gammln(xx)
c
c       Function used by Amoeba routine
c
      REAL gammln,xx
      INTEGER j
      DOUBLE PRECISION ser,stp,tmp,x,y,cof(6)
      SAVE cof,stp
      DATA cof,stp/76.18009172947146d0,-86.50532032941677d0,
     1  24.01409824083091d0,-1.231739572450155d0,.1208650973866179d-2,
     2  -.5395239384953d-5,2.5066282746310005d0/
      x=xx
      y=x
      tmp=x+5.5d0
      tmp=(x+0.5d0)*log(tmp)-tmp
      ser=1.000000000190015d0
```

```
      do 11 j=1,6
      y=y+1.d0
      ser=ser+cof(j)/y
11    continue
      gammln=tmp+log(stp*ser/x)
      return
      END
C  (C) Copr. 1986-92 Numerical Recipes Software 0Q-815='.
```