AN ABSTRACT OF THE THESIS OF

Jeffrey A. Longmate for the degree of Doctor of Philosophy in

Statistics presented on August 26, 1986.

Title: Analysis of Enumerative Data in Randomized Block Designs.

Redacted for Privacy

Abstract approved: ______________________________

Donald A. Pierce

Randomized block designs are often modeled by fixed treatment effects, an additive random effect that is shared by all observations in the same block, and a second random effect accounting for the residual variation. With enumerative data, the residual variation may be related to the treatment and block effects in a complex way. A two-stage model is presented, in which the parameter of a binomial or Poisson count is some one-to-one function of the outcome of the usual randomized block linear model. The implications of this model for analyses of variance are discussed, and some extensions of binomial and Poisson weighted least squares methods to overdispersed counts are adapted to the randomized block situation. Variations of the ANOVA and weighted least squares approaches are discussed and compared on a binomial and a Poisson example, and both approaches are compared to a direct maximum likelihood method on several simulated data sets.

Analysis of Enumerative Data
in Randomized Block Designs

by

Jeffrey A. Longmate

A THESIS

submitted to

Oregon State University

In partial fulfillment of
the requirements for the
degree of

Doctor of Philosophy

Completed August 26, 1986

Commencement June 1987

APPROVED:

Redacted for Privacy

Professor of Statistics in charge of major

Redacted for Privacy

Head of department of Statistics

Redacted for Privacy

Dean of Graduate School

Date thesis is presented: August 26, 1986.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

ANALYSIS OF ENUMERATIVE DATA IN RANDOMIZED BLOCK DESIGNS

1 INTRODUCTION

The matter addressed here is the analysis of counted responses from a randomized block experiment. The kinds of responses considered here might be thought of as Poisson or binomial data, except that block heterogeneity or treatment-by-block interactions may cause greater variation than that due to a Poisson or to a binomial distribution. Considering the fundamental nature of randomized block designs and the recent interest in generalized linear models, surprisingly little has been written about this situation.

Analysis of variance, often on a transformed scale, has long been used for enumerative data when random variation appears to be too large to attribute to a binomial or a Poisson distribution. Bartlett (1936) and Cochran (1943) mention randomized blocks in connection with this approach. Anscombe (1954) and Cochran (1954) noted that for analyzing counts, an analysis of variance on a transformed scale may be preferable to Fisher's maximum likelihood method when either (a) there is an extraneous source of variation making the binomial or Poisson maximum likelihood solution inappropriate, or (b) a non-iterative solution is desired for the sake of speed and simplicity. With the availability of powerful and flexible computer programs such as GLIM (Baker and Nelder 1978), speed and simplicity of computations are no longer very important, although conceptual simplicity is still of value. The general approach of Fisher's maximum likelihood method has, in recent years, been extended to deal with overdispersion by an approach which will be refered to as empirically weighted least squares (EWLS). The main objectives of this thesis are to point out the applicability of the EWLS approach to randomized block experiments, to compare it to the ANOVA approach, and to compare both to a direct maximum likelihood solution.

As an initial example consider a subset of the data from an experiment described by Cochran and Cox (1957, p46). Two soil fumigants were applied, each at two doses, in a two-by-two factorial design repeated in four fields. The response is the number of eelworms in a 400 gram soil sample. The data are given in Table 1 with the two fumigants designated CS and CK.

TABLE 1. Data from eelworm experiment

block:	1		2		3		4	
fumigant:	CS	CK	CS	CK	CS	CK	CS	CK
dose 1:	194	256	221	236	433	268	80	132
dose 2:	372	280	166	142	311	408	28	292

As will be seen below, the Poisson distribution does not adequately explain the deviation of these data from a model with additive treatment and block effects, even if a dose-by-fumigant interaction term is included. Cochran and Cox analyzed the full experiment as if the data were normal measurements with constant variance; however, it is likely that at least some component of variance in this experiment could be thought of as a Poisson-type variation.

It is probably uncommon for a model with purely Poisson error to be adequate for large counts. The coefficient of variation for a Poisson distribution with mean μ is $1/\sqrt{\mu}$, which is very small for data such as those in Table 1. Often there are other sources of random variation which are a larger fraction of the mean than this. Similar considerations hold for binomial data with large sample sizes.

Even with modest expected values, the variance of the counts may be too large to be explained by a binomial or Poisson error. The data in Table 2, made available through the courtesy of Mr. David Sewall, are counts of the number of responses out of 20 trials. They exhibit enough overdispersion to cast doubt on a binomial model. The experiment was done to study the effect of a fungicide used in orchards on the interaction between caterpillars and parasitic wasps. Blocks consisted of 40 caterpillars exposed over a short period to a colony of wasps. From each block 20 caterpillars were assigned to a control diet while the other twenty received one of two treated diets, in an incomplete block design. The response variable is the number of caterpillars in which a parasite developed to a certain stage.

TABLE 2. Data from caterpillar experiment. (n = 20)

	block					
treatment	1	2	3	4	5	6
control	12	8	15	15	17	17
treatment 1	4	1	0	-	-	-
treatment 2	-	-	-	0	7	7

The data given are for an experiment in which caterpillars were treated prior to exposure, so caterpillars from different treatment groups had to be kept in separate containers during exposure. Even though the caterpillars were randomly assigned to treatments, the grouped exposure may have generated a random effect shared by those insects receiving the same treatment. A similar experiment involving only post-exposure treatments lacked this clustering feature and fit a binomial model much better.

Whether the response is a count or a measurement, in a randomized block experiment a sample of blocks is obtained from some population, and subunits of each block, either natural or arbitrary, are randomly assigned to various treatments. When the response is a count, the sampling of blocks, assignment of treatments, and random environmental effects impinging on the experimental material, can be thought of as determining a response rate, p, for each treated subunit. But this response rate cannot be directly measured. Instead, a sample of n objects in the binomial case, or a sampled region of size n in the Poisson case, is obtained for each subunit and the sample yields a count, r, which determines an empirical response rate, $\hat{p} = r/n$. The binomial and Poisson errors will be termed sampling errors, even though in many experiments all of the experimental material is examined to obtain the counts. The residual error, i.e. the random effect accounting for the deviation of the counts from the values predicted by a model with additive treatment and block effects, depends on both the sampling error and the random perturbation of the response rate of the individual subunit. The latter is the overdispersion error relative to a binomial or Poisson model. A major complication when the response is a count is that the variances of the sampling error and of the overdispersion effect may depend in different ways on the (conditional) mean response rate, i.e. on the treatment and block effects.

In the case of normally distributed measurements, with several responses for each cell in the treatment-by-block layout, two sources of error in addition to block effects can in principle be identified. One source, sometimes called pure error, explains the variation among measurements in the same cell, that is, among the individual responses to the same treatment within a block. The other source, sometimes called treatment-by-block interaction, explains the deviations of the average responses for each cell from those predicted by a model with additive treatment and block effects. Both variances are usually assumed to be free parameters. In the common case with only one response per treatment within each block, these two sources of error are totally confounded, that is, not separately identifiable.

When the response is a count, two sources of residual error can be individually discerned even with only one count per cell. One source, the sampling error, is here assumed to be either binomial or Poisson. The other source, the overdispersion error, accounts for the variation in the local conditions by allowing for variation in the binomial or Poisson means in addition to that explainable by the block effect. Since the sampling variance is a known function of the mean, the excess variation over the sampling variance corresponding to the fitted means provides an estimate of the amount of overdispersion.

In the measured response setting with replication within cells, the treatment-by-block interaction, but not the pure error, provides an appropriate error estimate for inferences about treatment effects to a population of blocks. The separation of pure error from interaction is obviously important in that case. In the counted response setting, the variance of a contrast will depend on both the sampling variance and the overdispersion. The motivation for separating them comes from the fact that these two variances are not free parameters, hence the way in which the residual variance is related to the treatment and block effects will depend on the mixture of the two.

The sampling variance is a known function of the mean response rate. The overdispersion variance, while not generally a known function of the mean, may not be a free parameter either. When the expected counts are small, their non-negativity constrains the overdispersion variance. As a consequence, the overdispersion, like the sampling variance, may depend on the mean response, i.e. on the treatment and block effects, but possibly in a different way than the sampling variance.

Sometimes one or the other source of error is negligible. When such cases are identified, particularly simple methods become available. In other cases some assumption about the nature of the overdispersion may be reasonable. If a function determining the overdispersion variance in terms of the mean response can be assumed, an appropriate weighting scheme is suggested. A final reason for separating overdispersion and sampling error is that it suggests a two stage model which is useful for illustrating the difficulties that can arise when little can be assumed about the overdispersion. Among the issues that arise are the problems of estimating the components of variance for the two sources of error, of estimating the treatment effects when the variance of the data depends on those effects, and of assigning a standard error to the treatment effect estimates.

As previously mentioned, little has been written about these matters as they apply to the randomized block setting. There have been a number of papers dealing with overdispersion of enumerative data when the counts are independent. Although shared random block effects introduce a dependence between counts, these methods for independent counts provide a way of estimating treatment effects using weights that are conditional on block effects. Finney (1971) discusses the inflation of standard errors in a binomial regression to allow for variance in excess of the binomial. Kleinman (1973) presents an empirical weighting scheme for analyzing overdispersed binomial counts on the probability scale. Pierce and Sands (1975) translate Finney's suggestion from the probit setting to the logit setting, examine its performance when the overdispersion is homogeneous on the logit scale, and compare it to a very similar method motivated by the random logit model. Williams (1982) describes the approaches of Finney, and of Pierce and Sands, and gives instructions for implementing the methods in GLIM. Breslow (1984) discusses the Poisson case and also provides GLIM macros. Hinde(1982) uses GLIM to compute a maximum likelihood solution, assuming that the overdispersion error follows a normal distribution.

In addition to the literature on independent observations, there have been a number of papers on correlated counts; however, few of these are relevant to randomized block experiments. Several authors deal with cluster sampling (e.g. Altham 1976, Brier 1980, Cohen 1976) such as might be encountered in a retrospective study. A related situation occurs frequently in experiments involving littermates in which treatments are applied to entire litters. Haseman and Kupper (1979) discuss this type of study and give numerous references. It differs considerably from the randomized block experiment in that random litter effects are nested in treatments rather than crossed with them.

More to the point, Kleinman (1975) and Kempthorne (1982) discuss random block effects crossed with treatments, but both limit their discussion to the case of two treatments, and neither considers transformed scales. Zelen (1971) discusses testing for equality of relative risk before combining several two-by-two tables, but does not address the question of analysis when the relative risk is not constant. Hudson (1983) simulates randomized block experiments and compares the likelihood ratio chi-square test using differences of deviances to the approximate F-ratio analogous to the usual analysis of variance F-test. The conclusion is that the F-ratio is more robust to overdispersion than the difference of deviances, but still leads to excess rejections. Altham (1971) and Pierce (1976) both consider matched proportions, but emphasize matched Bernoulli trials, in which case overdispersion and sampling error, as defined here, are completely confounded.

There have been a number of generalizations of the tests of Cochran (1950) and Mantel and Haenszel (1959) for combining several contingency tables. Darroch (1981) provides a review. These all assume no three-way interaction of treatments, blocks and responses, i.e. no local overdispersion.

Of course randomized block experiments with counted responses have been analyzed in practice many times, with the most common approach probably being the usual randomized block analysis of variance, possibly after some transformation. For example both Cochran and Cox (1957) and Snedecor and Cochran (1980) introduce the randomized block design with examples of enumerative data, and both use ordinary randomized block ANOVA methods without comment.

2 THE MODEL

A two stage model can be used to describe a randomized block experiment. One stage consists of a linear model describing the randomized block structure of the experiment. It involves fixed treatment effects, random block effects, and random effects unique to each cell in the treatment-by-block layout. The totality of these effects determines, through some one-to-one function, the underlying rate of response for each cell, i.e., the binomial or Poisson mean for that cell. The second stage describes the sampling of n objects per cell in the binomial case, or of some subregion of each cell in the Poisson case, to obtain a count. Even though in many experiments all the material used is examined to obtain the counts, the binomial or Poisson errors will be generally referred to as sampling errors. The residual error - the random effect explaining the deviations of the individual counts from the values predicted from additive treatment and block effects - is a function of both the sampling error and of the first stage error unique to each observation. This latter will be called the overdispersion error.

As a basic model we will assume that, conditionally on the random effects described below which determine p_{ij} and μ_{ij}, the data are independent observations from a known distribution, either

$$r_{ij}|p_{ij} \sim \text{Bin}(n_{ij},p_{ij}) \tag{2.1}$$

or

$$(2.2) \qquad r_{ij}|p_{ij} \sim \text{Psn}(\mu_{ij}) \qquad \text{with } \mu_{ij} = n_{ij}p_{ij}$$

where $i = 1,\ldots,t$ indexes treatments, and $j = 1,\ldots,b$ indexes blocks. In both cases p_{ij} can be interpreted as a response rate, i.e., the expected count per unit of the denominator. It should be emphasized that n and p have different meanings in the two examples considered. In the binomial setting, p is a probability, while in the Poisson setting p may exceed one. The denominators, n, are integers in the binomial setting, but may be from a continuous scale in the Poisson setting. Expressing the mean count as np in both cases simply provides a convenient way of dealing with the possibility of imbalance in the denominators while directing inference to the mean response rates, i.e., the averages of the p_{ij} over the population of blocks. It is not meant to imply that the Poisson distribution is being considered as an approximation to the binomial. In the usual case where the Poisson denominators are all equal and perhaps only implicitly defined, the denominator can be assumed to be one without loss of generality.

The response rate, p, is assumed to be random, and for some specified one-to-one "link" function $\lambda(.)$,

$$(2.3) \qquad \lambda_{ij} = \lambda(p_{ij}) = x_i'\beta + b_j + u_{ij}$$

where for all i and j,

$$E(b_j) = E(u_{ij}) = 0,$$

$$V(b_j) = \sigma_b^2 \quad \text{and} \quad V(u_{ij}) = \sigma_u^2.$$

Here x_i' is a known row vector specifying the model for the treatment effects, β is a fixed vector of parameters, the b_j terms are random effects shared by members of the same block, and the u_{ij} represent random effects unique to each observation and independent of the b_j. All of the random terms are considered independent, and the block effect and unique terms are assumed to be realizations from separate homogeneous populations. Thus, in terms of the unconditional distribution of the data, r_{ij}, the terms b_j serve to introduce both overdispersion, relative to the binomial or Poisson variation, and positive dependence within blocks. The terms u_{ij} serve to introduce overdispersion not accounted for by blocking. In vector notation, writting $\lambda = (\lambda_{11},\dots,\lambda_{tb})'$,

$$E(\lambda) = X\beta$$

and

$$\mathrm{Var}(\lambda) = \sigma_b^2 BB' + \sigma_u^2 I,$$

where the row of X corresponding to λ_{ij} is x_i', B is a matrix indicating block membership, and I is the identity matrix.

The u_{ij} terms present somewhat of a nomenclature problem. The name treatment-by-block interaction, besides being unwieldy, is often used to refer to one possible contribution to u_{ij}. That phrase is also used to refer to the treatment-by-block interaction mean square in an analysis of variance table, which in this setting would estimate a function of both $\mathrm{Var}(u_{ij})$ and the sampling variance. A u_{ij} term will be called an overdispersion error. When it is necessary to distinguish it from the block effect, it will be called the local or unique overdispersion, since it affects an individual observation.

If there were no overdispersion, i.e. neither the b_j nor the u_{ij} were in the model, then we would have a generalized linear model (GLM). One advantage of generalized linear models over the usual ANOVA models is that GLMs do not assume constant variance, and additivity of fixed effects can hold on a transformed scale. But with the block and local overdispersion terms, the model for the linear predictor is the usual ANOVA model, and the problems of multiple assumptions on a single scale return.

That the random effects b_j and u_{ij} are additive on the same scale as the treatment effects is an assumption, with some rationale but not to be accepted uncritically; in fact, the consequences of the failure of this assumption will occasionally be discussed below. There is some appeal in taking homogeneous random effects to be additive on the same scale as the fixed effects, especially if the random effects might be due to the presence of unmeasured covariables that, if measured, would be included with the fixed effects. The form of the link, $\lambda(.)$, will not be discussed extensively here. To be definite, it will ordinarily be taken as the canonical form, i.e., logistic for the binomial, and logarithmic for the Poisson. These links have some advantages as summary statistics related to the fact pointed out by Törnqvist, Vartia and Vartia (1985) that log(a/b) is the only symmetric, additive, normed indicator of relative change. It should be noted that there is a somewhat novel aspect of the choice of link in this setting. If the block effects are not in fact additive on the scale used for the model, then they will serve to introduce additional local overdispersion.

Modeling the dependency between all counts in the same block by a common block effect is a simplifying assumption, but often a necessary one. The alternative of estimating separate covariances for each pair of treatments is only available when the number of blocks is large. The block effects are considered to be random because the interest here is in blocks that are sampled from (or at least thought to represent) some population. Block covariables and subpopulations will not be considered. It may at times be useful to consider the block effects to be fixed as a device for obtaining an analysis, but the local overdispersion term will always be considered random

The presence of the u_{ij} term with non-negligible variance is crucial in motivating the methods to be considered. If this term were absent, additivity of block effects would imply that a treatment contrast could be estimated as well by a single block with large denominators as by a large number of blocks. Then, if for purposes of inference about treatment effects we elect to treat the block effects as fixed, the data could be analyzed using well known binomial or Poisson maximum likelihood methods. Without a local overdispersion term however, it may be that no reasonably parsimonious model fits the data adequately.

One important thing that a local overdispersion term provides is the ability to attribute lack of fit to a random process rather than to inadequacy of the systematic part of the model. It may sometimes be more useful to fit simple models that imply large variances rather than more elaborate models, even when the latter is possible. This is sometimes done in classical randomized block experiments with a factorial treatment structure, when high order interactions among the treatments are assumed negligible, and pooled with the treatment by block sum of squares.

In addition to its role in accomodating lack of fit, the local overdispersion may be due to the assignment of treatments to the members of a block when blocks are not completely homogeneous. This appears to be the case in the caterpillar example. The data given are for an experiment involving a pre-exposure treatment that required keeping the treatment groups separated during their exposure to the wasps. As will be seen below, these data exhibit a substantial overdispersion. In a similar experiment in which more homogeneous blocks were possible, the residual variation was small enough to be explained by the binomial sampling distribution.

Overdispersion can be important even in designs without blocking. For example, if an observation is modeled as Poisson with mean μ, then the coefficient of variation is $1/\sqrt{\mu}$. Accepting the Poisson model for an arbitrarily large mean would mean accepting an arbitrarily small coefficient of variation. So when the counts are large we should expect an extra-Poisson source of variation. In the eelworm example, the size of the counts for which a Poisson model is reasonable will depend on the size of a region over which the spatial distribution of eelworms is nearly homogeneous. As n increases we expect additional sources of variation to become important.

In the binomial case, let $\hat{p} = r/n$. Then without any overdispersion, $\mathrm{Var}(\hat{p}) = p(1-p)/n$, which for fixed p approaches zero as n increases without bound. If the p_{ij} could somehow be measured directly rather than estimated from n_{ij} observations, they would probably still deviate from an additive model with treatment and block effects, implying an extra-binomial source of variation.

The sort of increase in n contemplated above involves an increase in the extent of the sample, in contrast to the sort of increase usually considered as motivation for the Poisson approximation to the binomial distribution. If counts were obtained by noting the presence or absence of eelworms in n very small subsamples of soil, there are two distinct procedures that might be used to increase n. Additional subsamples could be added, increasing the extent of the sample in the manner considered above, or the same sample could be divided into more subsamples, each of smaller size. The second method would result in the number of "presence" events approaching the number of worms in the sample. More formally, if np approaches μ as n increases, then the binomial count, r, converges in distribution to a Poisson random variable with mean μ. Is should be re-emphasized that throughout this thesis, n will refer to a measure of the extent of sampling of a region with a constant response rate, p, and that the use of the notation "np" for the Poisson mean is not meant to imply any relationship with a Binomial distribution.

To summarize, large denominators or large counts are cause for concern about overdispersion in any design. In randomized block designs, there are additional sources of overdispersion that may be important even with small denominators and small counts. These include block heterogeneity, interaction of treatments with blocks, and nonadditivity of block effects on the scale chosen for analysis. The overdispersion that may be present conditional on a block is modeled here by an additive, homogeneous local error term. Positive dependence within blocks is modeled by a random block effect, and the treatment means over the population of blocks are assumed to follow a linear model. The local overdispersion, block and fixed effect terms do not necessarily predict the response rate directly, rather the rate is some one-to-one function of the linear predictor, generally taken to be the inverse of a log or logit. The only sampling distributions we will consider are the Poisson and the binomial. These each involve an independence assumption which would have to be abandoned if underdispersion were to be modeled. But since it is overdispersion that is most frequently of concern in a randomized block experiment, and it can be modeled by the unique random effect u_{ij} in the linear predictor, these two sampling distributions will be adequate for most purposes.

Ignoring the linear predictor for the moment, the expectations and variances of r and $\hat{p}$ are summarized in Table 3. for the Poisson and the binomial cases. The moments are in terms of a random p, its mean π, and its variance which will generally be a function of π.

TABLE 3. Moments of r and $\hat{p}$ in terms of a random p

	Sampling Model	
Moment	$r\|p \sim \text{Psn}(np)$	$r\|p \sim \text{Bin}(n,p)$
$E(r\|p)$	np	np
$Var(r\|p)$	np	npq
$E(r)$	$n\pi$	$n\pi$
$Var[E(r\|p)]$	$n^2\sigma_\pi^2$	$n^2\sigma_\pi^2$
$E[Var(r\|p)]$	$n\pi$	$n[\pi - (\pi^2+\sigma_\pi^2)]$
$Var(r)$	$n[\pi + n\sigma_\pi^2]$	$n[\pi(1-\pi) + (n-1)\sigma_\pi^2]$
$E(\hat{p})$	π	π
$Var(\hat{p})$	$\pi/n + \sigma_\pi^2$	$\pi(1-\pi)/n + (1-1/n)\sigma_\pi^2$

NOTE: $\pi = E(p)$ and $\sigma_\pi^2 = Var(p)$, a function of π.

As n increases without bound, $Var(\hat{p})$ approaches Var(p) in both the binomial and the Poisson cases. In the binomial case, the minimum denominator is one. For n = 1, there is no overdispersion, the variance of $\hat{p}$ being its binomial variance. In the Poisson case, n may be continuous, so zero is its lower bound. As n approaches zero, Var(r) / Var(r | p = π) approaches one. Loosely put, as n approaches zero Var($\hat{p}$) approaches π/n, its value when r has an unconditional Poisson distribution.

If we consider the linear predictor, λ, for a single observation, the delta method can be used to get an approximate variance of p in terms of var(λ) = σ^2. Using a first order Taylor's approximation, $p(\lambda) \approx p(\mu) + \frac{dp}{d\lambda}(\lambda - \mu)$ where $\mu = E(\lambda)$. So $Var(p) \approx (\frac{dp}{d\lambda})^2\sigma^2$ where the derivative is evaluated at $\mu \approx \lambda(\pi)$. In the Poisson case with a log link, $p(\lambda) = \exp(\lambda)$, so $\frac{dp}{d\lambda}$ evaluated at $\lambda(\pi)$ is π, and the delta method approximation is $Var(p) \approx \pi^2\sigma^2$. Substituting this for σ_π^2 gives in the Poisson case

(2.4) $$Var(r) \approx n\pi + n^2\pi^2\sigma^2$$

and

(2.5) $$Var(\hat{p}) \approx \frac{\pi}{n} + \pi^2\sigma^2.$$

In the binomial case with a logit link, $p(\lambda) = [1 + \exp(-\lambda)]^{-1}$, and $\frac{dp}{d\lambda}$ evaluated at $\lambda(\pi)$ is $\pi(1 - \pi)$, so $Var(p) \approx \pi^2(1 - \pi)^2\sigma^2$. Substituting for σ_π^2 gives

(2.6) $$Var(r) \approx n[\pi(1 - \pi) + \pi^2(1 - \pi)^2(n - 1)\sigma^2]$$

and

(2.7) $$Var(\hat{p}) \approx \pi(1 - \pi)/n + \pi^2(1 - \pi)^2(1 - 1/n)\sigma^2.$$

Equations (2.5) and (2.7) display the essential problem here. The first term is often fairly well known once π is estimated. The second term depends on σ^2 which is an unknown, free parameter. As π varies, the two terms do not remain proportional to one another, causing a variance heterogeneity that cannot be removed by a single transformation for all values of σ^2.

In an ordinary randomized-block analysis of variance problem, the analog of the entire quantity $\pi/n + \pi^2\sigma^2$ is a constant which is estimated directly from the treatment-by-block interaction mean square. With enumerative data, the situation is complicated by the non-proportionality of the two terms in the variance, and by the fact that estimates of π provide information about the first term that is often worth capitalizing on.

3 METHODS

Three general approaches to the analysis will be considered: ordinary (unweighted) randomized block analysis of variance, possibly after some transformation of the data; empirically weighted least squares; and direct maximum likelihood. Non-parametric approaches will not be discussed in detail, as these are mainly oriented toward hypothesis testing. There are of course many possible variations within each of these approaches, and, as with any classification, the boundaries may sometimes be fuzzy.

3.1 The ANOVA approach

In the ANOVA approach, three scales are of particular interest: (i) the original scale of counts, r, or empirical rates, $\hat{p} = r/n$; (ii) the scale that stabilizes the sampling variance, i.e. the square root transformation in the Poisson case, and the angular transformation in the binomial case; and (iii) the scale on which the linear model is assumed, here taken to be either the log or the logit scale. Choice among these scales may depend on the range and magnitude of the counts, and suppositions about the sources of error.

By the ANOVA approach is meant an ordinary randomized block analysis of variance, possibly after transforming the data, using the treatment-by-block interaction mean square as an error term. It is useful to think of the treatment-by-block interaction sum of squares as composed of contributions from orthogonal contrasts. In complete block designs, the standard error of a treatment contrast can in principle be estimated very directly by computing the value of the contrast for each block. However if the number, b, of blocks is small, this method gives only b-1 degrees of freedom for estimating the standard error. If the transformed counts have constant variance, the standard errors of orthogonal contrasts could be pooled into a treatment-by-block interaction sum of squares with (t-1)(b-1) degrees of freedom, where t is the number of treatments. So if there are three or more treatments, i.e. if any pooling at all is possible, the degrees of freedom available for estimating the constant variance will be at least double the number available for estimating the error of a single contrast without pooling.

Since the justification for pooling rests on the assumption of constant variance, satisfying this assumption becomes a major goal in choosing a transformation when the number of blocks is modest. But unless the overdispersion and the sampling distributions have similar mean-variance relationships, there may be no transformation that simultaneously stabilizes both sources of error, and a transformation that could stabilize their combined effect would depend on knowledge of their relative magnitude. For example, the Poisson distribution with mean μ has variance μ. But if we believe that the overdispersion has a roughly constant coefficient of variation, then its variance must be roughly proportional to μ^2. The variance of a count is then approximately

$$\text{(3.1)} \qquad \mathrm{Var}(r) \approx \mu + \mu^2\sigma^2$$

for some σ^2. The square root transformation then stabilizes the Poisson portion of the variance, i.e. that part proportional to μ, but not the extra-Poisson variation. It will only be effective if σ^2 is relatively small. Bartlett (1947) gives the transformation that stabilizes the variance when σ^2 is known. Some calculations will be discussed in the next chapter that may help decide when σ^2 is small enough to be neglected.

An additional problem with the square root transformation, as well as with the arcsine transformation for binomial data, is that if the fixed effects of equation (2.3) are to be estimated assuming a canonical link, the contrasts of transformed counts are not the most appropriate estimators of treatment effects, even if they do have constant variance. There has been some discussion in the literature of whether additivity or constant variance ought to be the main objective of using a transformation with an analysis of variance (Bartlett 1936, 1947, 1954; Cochran 1947; Tukey 1949; Fisher 1954) If the treatments have no structure, the only additivity of concern is the additivity of the block effects with the treatment effects. If these are not additive on the transformed scale, e.g. if a treatment effect is larger in those blocks that have large responses for the control, the resulting "systematic" lack of fit will contribute to the residual sum of squares and inflate the variance estimate. In addition, extrapolating results to a different population of blocks will be even more hazardous than usual. If, as described above, the size of a treatment effect tended to increase with the block effect, its estimate could not be applied to a subpopulation with, for example, small block effects. What is really at issue here is the distinction between systematic and random variation. Without block covariables there is little basis for separating the two.

If the treatments have, for example, a factorial structure, additivity means that the effects of one factor at a particuar level of another generalize to all levels of the second factor. If a nonlinear transformation is made to a different scale then the constant effect on the additive scale translates into different effects at different levels of the second factor. If terms are not introduced into the model to allow interaction of factors, nonadditivity of fixed effects will also inflate the residual sum of squares. Failure of the additivity assumption may be serious for estimation, but is less so when interest centers on testing a null hypothesis, since the absence of a treatment effect is insensitive to the choice of the link or transformation.

If the overdispersion was the dominant source of error, one might consider using the link function as a transformation. It would be the approximate variance stabilizing transformation, and it would be the correct transformation for preserving additive effects. However, the use of the link function as a transformation relies heavily on the adequacy of the assumed model.

Variance stabilizing transformations correct for the differences in variance between counts with different expectations. They do not correct for unequal variances due to unequal denominators. Cochran (1943) discusses appropriate weights when the denominators differ substantially. In the discussion of the ANOVA approach the equal denominator case will be emphasized since that is the situation where the distinction between the ANOVA and the EWLS approach is sharpest.

3.2 Empirically weighted least squares approach

As mentioned above, without the random block and local overdispersion effects, the model introduced in chapter 2 would become an instance of a generalized linear model as described by Nelder and Wedderburn (1972). The maximum likelihood solution could then be calculated by iterative weighted least squares, e.g. by Fisher's scoring method. Several extensions to this approach have been proposed for dealing with a model with a local overdispersion term, but without a random block effect. These will be refered to as empirically weighted least squares methods, after Kleinman (1973). These methods can be used to analyze randomized block designs by treating the block effects as fixed. The rationale for fixing block effects will be deferred to a subsequent chapter where the empirically weighted least squares methods are discussed in more detail. The general idea is that block effects largely cancel out of treatment contrasts, so whether they are considered fixed or random is not critical to the assessment of the standard errors of contrasts. This is not to say that the blocks are really to be thought of as fixed. Treating them as fixed is simply a technical means to an approximation. In particular the local overdispersion error, which includes any block-by-treatment interaction, is still to be treated as random. It should be noted that the ANOVA approach can be thought of as fitting block effects by least squares, as if they were fixed effects, in order to obtain sums of squares and treatment effect estimates.

If we knew the variance of the overdispersion term as a function of its mean, i.e. if we knew σ_{π}^2 in the bottom line of Table 3, we could in principle devise an approximate variance stabilizing transformation by the same method used to obtain the square root and angular transformations when the variance is entirely Poisson or binomial. Bartlett (1947) discusses such a transformation in the Poisson setting. An alternative would be to use the knowledge of the resulting mean-variance relationship given the treatment and block effects, to estimate a variance for each count which could then be used in a weighted least squares fit. For example, if the variance of a count was given by equation (3.1) and σ^2 was known, we could use a weighted regression with weights $1/(\hat{\mu} + \hat{\mu}^2\sigma^2)$. The empirically weighted least squares approach is based on this idea, with the difference that only the form of the overdispersion variance function is assumed. The value of σ^2 is estimated from the data.

Since the estimate of σ^2 both depends on the estimates of response rates and influences them by determining the weights, an iterative approach can be taken. An initial set of variance components can be assumed, then re-estimated in light of the resulting weighted estimates of the means, providing new variance components for the next fit. This approach includes a variety of methods, at least one for each assumption about the mean-variance relationship of the overdispersion.

To summarize, in the ANOVA approach we deal with the mean-variance relationship by using a transformation, unweighted means, and empirical variances that do not involve the means. The transformation is usually chosen *a priori*. In the empirically weighted least squares approach, we deal with the mean-variance relationship by using a weighted analysis. The actual weighting function is not assumed *a priori*, although a parametrized family of functions such as that determined by equation (2.4) is assumed. Actual weights are determined by iterating between estimating means using assumed weights, and estimating new weights using the estimated means and deviations. Standard error estimates will generally involve the estimated means. Both approaches may involve assumptions about the mean-variance relationship of the overdispersion. An advantage of the weighted least squares approach is that it does not require that the overdispersion and sampling variances be proportional when neither is negligible.

Since the weighted least squares approach seems to be a refinement of the unweighted least squares implicit in the ANOVA calculations, it might seem like the better approach to take in general. However, the ANOVA approach has some particular merits. Among these are its familiarity to workers in a variety of fields, and its light computational burden. In addition, the error term for treatment effects will generally encompass both the sampling variance and the overdispersion automatically. Some versions of the EWLS approach start as binomial or Poisson maximum likelihood analyses and require the data analyst to check the lack-of-fit. A completely specified algorithm for such a method requires a rule specifying the amount of evidence of overdispersion necessary to invoke empirical weighting.

If a transformation was chosen based on the assumption of binomial error when in fact there was substantial overdispersion with constant variance on the logit scale, the efficiency of ANOVA estimators might suffer, but their imputed standard errors will probably be reasonable. If a weighted least squares analysis is done assuming binomial error, the usual logistic regression would lead to standard errors based on the fitted means rather than the observed lack of fit. Unless we apply some adjustment based on an empirical estimate of the amount of overdispersion, the standard errors could be grossly underestimated. The simplest empirically weighted least squares method amounts to just such an adjustment.

In some circumstances the fact that the ANOVA approach uses unweighted means may be an advantage. Appropriately weighted means are more efficient estimators of additive effects than unweighted means, but if the effects being estimated interact with other factors, perhaps because no simple additive scale can be found, unweighted means may be preferable when the other factors are representative of some population. If a treatment interacts with another treatment, but the interaction is modest, one might still be interested in the main effects as summaries. A weighted analysis would emphasize the effect of one treatment at the levels of the other treatment that result in small variance. If a treatment interacts with blocks, the interaction cannot be modeled without either completely saturating the model with parameters or introducing block covariates. A weighted mean in this case will be biased toward the values for the blocks giving the smallest variances. An unweighted mean will have a larger variance, but it will be unbiased for the mean treatment effect for the population of blocks, provided that the blocks are a random sample.

3.3 Direct maximum likelihood

The direct maximum likelihood approach will be treated more as a standard against which the other methods can be compared in simulations than as a serious suggestion for data analysis. If we assume that in the model for the linear predictor given by equation (2.3) the block and local overdispersion effects have normal distributions, then the likelihood can be written in terms of nested integrals involving the normal density and the density of the sampling distribution. These can be evaluated by repeated quadratures, and maximized by various schemes for searching in the parameter space.

A program was written to do the maximum likelihood computations in the binomial case. It uses Gaussian quadrature to evaluate the integrals and uses the Davidon-Fletcher-Powell (DFP) algorithm as described by Reklaitis, Ravindran, and Ragsdell (1983, p114) to search for the maximum likelihood. The algorithm starts as a gradient ascent method, using only first derivatives. As the maximum is approached, it estimates the second derivatives from the changes in the first derivatives and gradually shifts to an approximate Gauss-Newton method. The direct maximum likelihood approach involves much more computation than the iterative least squares approach. The Davidon-Fletcher-Powell algorithm was used instead of a Gauss-Newton procedure in order to avoid the need to compute second derivatives until the solution was found.

Interest was centered on inference about a null effect in the presence of a substantial additive nuisance effect, and simulated data for this situation were generated. No attempt was made to do a detailed Monte Carlo analysis of the methods compared. Rather a few trials were generated at each of a few combinations of variance component values in order to get an idea of how the methods compared on individual data sets over a range of conditions.

3.4 Other methods

Various rank and randomization based methods have been proposed for randomized block designs. These are oriented toward hypothesis testing, but confidence intervals could be placed around a shift parameter on some suitable scale. This may seem somewhat unnatural, since a smooth shift would create noninteger counts. Some sense could be made of the shift by treating it as a shift of the underlying rates, assuming that the counts are those most likely according to the sampling distribution.

A problem with a randomization test is that it may have very low power if there are few blocks. In the caterpillar example, the randomization test would be a sign test with only three pairs, hence with a minimum possible significance probability of 0.125. However, in related experiments with post-exposure treatments, the randomization of individual caterpillars to treatments could be used, creating many more possible values of a test statistic and hence a lower minimum p-value. If there is a possibility of treatments and blocks interacting, the test using the allocation of individuals may provide evidence against the strong hypothesis of no treatment effect in any block, but it can not provide a valid significance level applicable to the weaker hypothesis that the mean response rates are the same for the two treatments despite the possibility of some blocks responding more strongly to one treatment and some blocks to the other. Such an interaction may be hard to imagine in the caterpillar example, but in the experiment for which data were given, i.e. the one with a pre-exposure treatment, the grouped exposure to the colony of parasites raises a serious objection to the use of the unrestricted randomization distribution.

Using a parametric approach such as an EWLS method, the difference in the pre-exposure and postexposure experiments might lead us to expect different amounts of overdispersion, but we would base the significance of our statistic on the variance components estimated from the data, not just on the randomization scheme. In addition, a parametric approach allows variance estimates to be pooled for the blocks comparing different treatments and it allows arbitrarily defined contrasts to be estimated.

4. ANOVA APPROACH

4.1 Fundamentals

Perhaps the oldest way of handling enumerative data from a randomized block experiment is to use an ordinary randomized block analysis of variance with the block-by-treatment interaction used as the error term for computing the standard errors of treatment contrasts. The use of such an error term amounts to pooling the (b-1) degree of freedom variance estimates for a set of orthogonal treatment contrasts. The essential features of what is meant here by the ANOVA approach are the use of unweighted means to estimate the treatment effects, and the use of a pooled estimate of a supposedly constant variance to estimate their standard errors. Pooling is important when the number of blocks is small. If the range of counts is large, the variance will not be constant so pooling will not be appropriate. A transformation may then be applied to the data in an attempt to stabilize the variance.

Suppose we have t treatments, each applied once in each of b blocks. Let y_{ij} denote the response for treatment i in block j, possibly after some transformation. Let $L_1, \ldots, L_{t-1}$ be mutually orthogonal contrasts where $L_k = \Sigma_i \lambda_{ki} \bar{y}_{i \cdot}$, and where $\bar{y}_{i \cdot}$ denotes the mean over blocks of the transformed counts for treatment i. One way to estimate the variance of a contrast in a randomized block design is to compute the value of the contrast in each block and use the sample variance of that particular contrast. If the number of blocks is large, this method can supply an adequate estimate with a minimum of assumptions. However, if the number of blocks is small, this method provides only b-1 degrees of freedom for estimating the variance. If there are three or more treatments, we can do substantially better if we can pool the estimates from different contrasts.

Assuming that the residual errors are independent with constant variance, the variance of a contrast is $V(L) = (\Sigma\ \lambda^2)\sigma^2/b$ where σ^2 is the residual variance, i.e. the variance of the sum of all random effects except the block effects. If we let s_k^2 denote the sample variance of L_k, we can estimate σ^2 by $\hat{\sigma}_k^2 = (b/\Sigma_i \lambda_{ki}^2)s_k^2$. Each $\hat{\sigma}_k^2$ is independent so we can form the pooled estimate $\hat{\sigma}^2 = \Sigma\ \hat{\sigma}_k^2/(t-1)$ which has $(t-1)(b-1)$ degrees of freedom.

4.2 Difficulties with counted responses

Because the pooling discussed above is only clearly justifiable if the residual variance is the same for all responses, and the variances of counts will generally differ if the range of the data is large due to treatment effects, a transformation is often needed. A difficulty is that the usual variance stabilizing transformations, the square root in the Poisson case and the angular transformation in the binomial case, only stabilize the sampling variance which is just one source of the residual variation. If sampling variance is the dominant source, the usual transformations are adequate. However, if the overdispersion on the canonical link scale is the dominant source, the link function may do a better job of stabilizing the variance. If neither the sampling variance nor the overdispersion dominates, then neither class of transformation will stabilize the variance over a wide range of means, though either may be an improvement over no transformation.

An added difficulty with the square root or the angular transformation is that with a linear model on the canonical link scale, these transformations will spoil additivity. The link transformation will give approximately additive effects provided that the supposed model is correct. Of course the data often provide little means of distinguishing between alternative choices of a link. Because the mean-variance relationship of the residual error results from the combined effect of sampling and overdispersion errors, some other transformation may do a better job of stabilizing the variance than either the traditional variance-stabilizing transformation or the link transformation. Bartlett (1947) gives a compromize transformation in the Poisson case that depends on the relative contributions from Poisson and extra-Poisson variation. Box and Cox (1964) and Draper and Hunter(1969) have described an empirical approach to the choice of a transformation that seeks to find a compromize between several criteria such as constant variance, additivity of fixed effects and power for testing fixed effects. Their method involves repeatedly trying different power transformations while keeping track of various diagnostic statistics.

The remainder of this chapter will be devoted to an examination of three related matters. One is the difficulty of simultaneously stabilizing both the sampling variance and the overdispersion variance. The second is an examination of approximate variance components for a contrast with an eye toward identifying situations in which one component is negligible. The third is a brief consideration of a compromise between assuming constant variance for all counts and using only b-1 degrees of freedom for each contrast.

4.3 Stabilizing two variances

Focusing on the binomial case for the moment, equation (2.7) can be generalized to give $\mathrm{Var}(\hat{p}) \approx \pi(1-\pi)/n + (\frac{dp}{d\lambda})^2(1-1/n)\mathrm{Var}(\lambda)$. Writing y(.) for the transformation to be applied to $\hat{p}$ and approximating it by another Taylors expansion about π gives

$$(4.1) \qquad \mathrm{Var}(y(\hat{p})) \approx (\frac{dy}{dp})^2[\pi(1-\pi)/n + (\frac{dp}{d\lambda})^2(1-1/n)\mathrm{Var}(\lambda)]$$

where the derivatives are evaluated at the expansion points. When $\mathrm{Var}(\lambda) = 0$, setting $\mathrm{Var}(y(\hat{p}))$ equal to a constant gives $y_1(\hat{p}) = \mathrm{arcsine}(2\hat{p} - 1)$ and $y_2(\hat{p}) = \mathrm{arcsine}(\sqrt{\hat{p}})$ as solutions, the usual approximate variance stabilizing transformations. These are related by $y_1 = 2y_2 - \pi/2$. The latter is the more commonly encountered transformation, but the former seems the easier function to visualize, and has a variance of approxiamtely 1/n.

As n increases for fixed π, the overdispersion dominates and a similar argument leads to $y = \lambda$ as an approximate variance stabilizing transformation. The use of λ as a variance stabilizing transformation is probably not as satisfactory in practice, since we would typically have considerable doubt as to the scale with constant overdispersion variance. When neither term dominates, the variance can still be made constant by the angular transformation if λ is a linear function of y, but not for arbitrary λ. So essentially the only case when an angular transformation stabilizes variance regardless of the amount of overdispersion or the size of n, is when the overdispersion is also homogeneous on the angular scale.

Table 4 gives the variance of the transformed counts given the block effect in terms of the overdispersion variance, σ_u^2 and of $E(p|b) = \pi$. Variances are given in the two situations of interest for both the usual variance stabilizing transformation and the link transformation.

TABLE 4. Variance of transformed counts.

BINOMIAL WITH LOGIT LINK: $p = 1/(1 + \exp(-\lambda))$

$y(\hat{p})$	Var(y)
$\text{arcsine}(2\hat{p} - 1)$	$1/n + \pi(1-\pi)(1-1/n)\sigma_u^2$
$\log(\hat{p}/(1 - \hat{p}))$	$1/(n\pi(1-\pi)) + (1-1/n)\sigma_u^2$

POISSON WITH LOG LINK: $p = \exp(\lambda)$

$y(\hat{p})$	Var(y)
$2\sqrt{\hat{p}}$	$1 + n\pi\sigma_u^2$
$\log(\hat{p})$	$\pi + n^2\sigma_u^2$

In the binomial case with the linear model on the logit scale, the angular transformation stabilizes the binomial variance but not the overdispersion, while the logit transformation stabilizes the overdispersion but not the binomial variance. In the Poisson case with the linear model on the log scale, the square root transformation stabilizes the Poisson variance but not the extra-Poisson variance, while the log transformation stabilizes the overdispersion but not the Poisson variance.

Although assuming homogeneous variance on the scale of the canonical link prevents exact stabilization of variance via the traditional transformation, we might still ask under what conditions the usual transformation might be good enough. The obvious answer is when the overdispersion is negligible. In the next section we examine the effects of both block and local overdispersion on the variance of a contrast when the angular transformation is used despite the linear model being on the logit scale.

4.4 Variance components for a contrast

If we consider the variance of a contrast of transformed proportions we can identify three variance components. The variance of the conditional binomial or Poisson distribution will contribute to the variance of the contrast, as will the local overdispersion. In addition, if the block effects are additive on one scale and we transform to another, some of the block variance will contribute to the local error. In evaluating the merits of the ANOVA approach, and deciding on an appropriate transformation it is desirable to have some idea of the relative magnitude of these sources of variance.

Let $L = \Sigma_i \lambda_i y(\hat{p}_{ij})$ be a contrast of transformed proportions in block j, where $\Sigma_i \lambda_i = 0$. Then

$$\text{Var}(L_j) = \text{Var}[E(L_j|b_j)] + E[\text{Var}(L_j|b_j)]. \qquad (4.2)$$

If the transformation, y, yielded additive block effects, $E(L_j|b_j)$ would not depend on b_j and its variance would be zero. So the first term in the decomposition of $\text{Var}(L_j)$ is a measure of the amount that non-additivity of the block effects contributes to the contrast variance. The second term measures the contribution from the sources of variation impinging on individual observations. It can be further decomposed into contributions from the sampling distribution, and the local overdispersion.

Let $y(\hat{p}) = \text{arcsine}(2\hat{p} - 1) \approx y(p^{o}) + (\hat{p} - p^{o})/(p^{o}q^{o})^{1/2}$ for $\hat{p}$ near p^{o}, where $q^{o} = 1 - p^{o}$. Let $p(\lambda) = 1/(1 + \exp(-\lambda))$, which for λ near λ^{o}, can be approximated by $p(\lambda^{o}) + p^{o}q^{o}(\lambda - \lambda^{o})$. Then for small σ_{u}^{2},

$$\text{Var}(L_{j}|b_{j}) = \Sigma_{i}\ \lambda_{i}^{2}\text{Var}(y(\hat{p}_{ij})|b_{j})$$

$$\approx \Sigma_{i}\ \lambda^{2}[1/n_{ij} + p_{ij}^{o}q_{ij}^{o}(1 - 1/n_{ij})\sigma_{u}^{2}]$$

where $p_{ij}^{o} = p(x_{i}'\beta + b_{j})$.

For small σ_{b}^{2},

$$E(p_{ij}^{o}q_{ij}^{o}) \approx p_{i}^{o}q_{i}^{o} - (p_{i}^{o}q_{i}^{o})^{2}\sigma_{b}^{2},$$

where $p_{i}^{o} = p(x_{i}'\beta)$ so

$E[\text{Var}(L_{j}|b_{j})] \approx \Sigma_{i}\ \lambda_{i}^{2}[1/n + p_{i}^{o}q_{i}^{o}\{1 - (p_{i}^{o}q_{i}^{o})\}\sigma_{b}^{2}(1 - 1/n)\sigma_{u}^{2}]$.

For $\sigma_{b}^{2} < .1$, the non-linearity correction term $\{1 - (p^{o}q^{o})\}\sigma_{b}^{2} > 0.975$. For $\sigma_{b}^{2} < 1.0$ the term is always greater than 0.75. Since only a very crude approximation is sought here and 1.0 is the largest variance we will consider, we can neglect the correction. Similarly we can neglect the (1 - 1/n) term if n is large, so

$$E[\text{Var}(L_{j}|b_{j})] \approx \Sigma_{i}\ \lambda_{i}^{2}[1/n + p_{i}^{o}q_{i}^{o}\ \sigma_{u}^{2}]$$

This can be further decomposed into a binomial component:

$$\Sigma_{i}\ \lambda_{i}^{2}/n_{ij}$$

and an extra-binomial component:

$$\Sigma_{i}\ \lambda_{i}^{2}[p_{ij}^{o}q_{ij}^{o}\sigma_{u}^{2}].$$

Returning to the first term in equation (4.2),

$$E(L_j|b_j) = \Sigma_i \; \lambda_i E(y(\hat{p}_{ij})|b_j) \approx \Sigma_i \; \lambda_i y(\overset{o}{p}_{ij}).$$

Taking first order Taylor's approximations to both y and p gives

$$E(L_j|b_j) \approx \Sigma_i \; \lambda_i [y(\overset{o}{p}_i) + b_j(\overset{o}{p}_i \overset{o}{q}_i)^{1/2}]$$

and

$$\mathrm{Var}[E(L_j|b_j)] \approx [\Sigma_i \; \lambda_i (\overset{o}{p}_i \overset{o}{q}_i)^{1/2}]^2 \sigma_b^2.$$

To summarize:

$$\mathrm{Var}(L_j) \approx \left\{ \begin{array}{ll} \mathrm{Var}[E(L_j|b_j)] \approx & [\Sigma_i \; \lambda_i (\overset{o}{p}_i \overset{o}{q}_i)^{1/2}]^2 \sigma_b^2 \\ & \text{non-additivity component} \\ \qquad + & \left\{ \begin{array}{l} \Sigma_i \; \lambda_i^2/n_{ij} \\ \text{binomial component} \\ \quad + \\ \Sigma_i \; \lambda_i^2 \overset{o}{p}_i \overset{o}{q}_i \sigma_u^2 \end{array} \right. \\ E[\mathrm{Var}(L_j|b_j)] \approx & \end{array} \right.$$

extra-binomial component.

These are all very rough approximations, but the aim here is to get an idea of when each of the three sources of error is likely to be important.

Consider two treatments, and let $L = y(p_2) - y(p_1)$, where y is the angular transformation. Table 5 compares the non-additivity and overdispersion components for several pairs of probabilities. λ_1 and λ_2 denote the logits of p_1 and p_2 respectively. The fourth column is the relative nonadditivity, i.e. the nonadditivity component of Var(L) divided by σ_b^2. The fifth column is the extra-binomial variance component divided by σ_u^2. The final column is their ratio which may be interpreted as the multiple by which the block variance must exceed the overdispersion variance for non-additivity due to link misspecification to be roughly comparable to the overdispersion. Since the angular transformation is being used, the binomial variance component is 2/n regardless of p.

Table 5. Comparison of Non-additivity and Extra-binomial Variance Components

p_1	p_2	$(\lambda_1 - \lambda_2)$	non-add/σ_b^2	x-bin/σ_u^2	ratio
.5	.4	.406	.0001	.49	4900
.1	.07	.406	.0022	.15	70
.5	.1	2.2	.04	.34	9
.1	.012	2.2	.036	.10	3

From Table 5 we see immediately that the non-additivity arising from the sort of link misspecification considered here is only likely to be of any concern if treatment effects are quite large. Clearly a null hypothesis of no treatment effect is insensitive to the choice of link. Table 5 provides a rough guide to how that invariance fails as treatment effects increase.

To compare the binomial error with the other sources, it is necessary to have some idea of what constitutes a large variance on the logit scale. Table 6 gives the range of probabilities corresponding to a range of four standard deviations on the logit scale, centered on the logit of 0.5 or 0.1.

Table 6. Probability Intervals

σ^2	σ	$p(\lambda(\pi) +/- 2\sigma)$	
		$\pi = 0.5$	$\pi = 0.1$
1.0	1.0	(.119 , .881)	(.015 , .450)
.1	.316	(.347 , .653)	(.056 , .173)
.01	.1	(.450 , .550)	(.083 , .120)
.001	.0316	(.482 , .516)	(.095 , .106)

Using n = 40 and the rather large values of $\sigma_b^2 = 1.0$ and $\sigma_u^2 = .3$ we get the approximate variance components listed in Table 7.

Table 7. Components of Variance for a Contrast
($\sigma_b^2 = 1.0$, $\sigma_u^2 = 0.3$, n = 40)

p_1	p_2	bin.	x-bin.	non-add	V(L)
.5	.4	.05	.147	.001	.198
.1	.07	.05	.045	.002	.097
.5	.1	.05	.1	.04	.19
.1	.012	.05	.03	.036	.116

Although the approximations used are not particularly good for such large variances there is a suggestion of a substantial mean-variance dependency despite the use of the angular transformation. If, in an application, *a priori* considerations allow a reasonable upper bouund to be placed on the random effect variances, some similar calculations might be helpful in deciding if overdispersion or non-additivity contribute enough to the variance of contrasts to be of any concern.

For example, using the more modest values of $\sigma_b^2 = .1$ and $\sigma_u^2 = .01$, we get V(L) = .05 to one significant digit for all the probabilities in the table. However, using n = 400 we get the values summarized in Table 8.

Table 8. Components of Variance for a Contrast ($\sigma_b^2 = .1$, $\sigma_u^2 = .01$, n = 400)

p_1	p_2	bin.	x-bin.	non-add.	V(L)
.5	.4	.004	.0049	.00001	.0089
.1	.07	.004	.0015	.00022	.0057
.5	.1	.004	.0034	.004	.0114
.1	.012	.004	.0010	.0036	.0086

With such a large n there is still a sizeable difference in variances of the contrasts for the different levels of the nuisance effect.

4.5 A pooling compromise

If the variance cannot be made constant by the use of a transformation, perhaps because neither the sampling error nor the overdispersion can be neglected, or perhaps because a different transformation is needed to make the fixed effects additive, then standard errors of each contrast could be separately estimated. This would avoid pooling variance estimates from contrasts whose variances differ due to substantially different fixed effects. Each such t-statistic would have b-1 degrees of freedom. If the treatments have a factorial structure we can often do better. Consider for example, a factorial experiment where treatment T1 has t1 levels, and treatment T2 has t2 levels. Suppose that the effect of T2 is large, and interest centers on the more modest effect of T1. Let t = t1xt2. The usual decomposition of treatment-by-block interaction is:

source	df
TxB	(t-1)(b-1)
T1xB	(t1-1)(b-1)
T2xB	(t2-1)(b-1)
T1xT2xB	(t1-1)(t2-1)(b-1)

This decomposition may be useful if one treatment interacts with blocks to a much greater extent than the other. The T1xB mean square can be used to form estimates of the variance of contrasts in the levels of T1 averaged over the levels of T2. If the three mean squares making up the treatment-by-block interaction do not differ too much, and if separate estimates of contrasts within the levels of T2 are of interest, the following decomposition might be useful.

source	df
TxB	(t-1)(b-1)
T1xB|T2=t1	(t1-1)(b-1)
...	...
T1xB|T2=t2	(t1-1)(b-1)
T2xB	(t2-1)(b-1)

The mean squares with (t1-1)(b-1) degrees of freedom provide estimates of the residual variance appropriate for inference about all t1-1 contrasts of the levels of T1 at a specified level of the nuisance factor, T2. This is a compromise in that we are pooling estimates of variance for those counts whose means are close due to a common value of the factor with the large effect.

Two further steps could be taken. The variance estimates for a contrast at different levels of the nuisance treatment might be smoothed somehow, and if additivity can be assumed, the corresponding estimates might be combined in a weighted mean. Smoothing the variances could be accomplished by regressing them on some *a priori* predicted values such as the approximation to the variance of a contrast used in the last column of Table 7. A simpler procedure, involving fewer assumptions as well as fewer computations, would be to use their isotonic regression (Barlow, Bartholomew, Bremner, and Brunk 1972, ch.1). Forming a weighted mean would essentially pool t2(t1 - 1)(b - 1) degrees of freedom. It would be a compromise between ANOVA and EWLS in that it would weight according to fixed effects, but not according to block effects. It will be argued in section 5.5 that given additivity, substantial block effects, and modest residual errors, observations on the same treatment in different blocks should be weighted according to their precision. The sort of *ad hoc* procedure just outlined may still be attractive because it consists of a fairly natural progression of comprehensible stages governed by decisions about the possible assumptions. If the assumption that the residual variance is independent of the treatment is made, we progress to the alternative breakdown of the treatment-by-block interaction. If constant variance is assumed, we can pool the variance. If variances appear heterogeneous some other sort of smoothing can be used. And if we are willing to assume additivity after examining the estimates at different levels of the nuisance factor, we can combine the estimates.

5 WEIGHTED LEAST SQUARES

Empirically weighted least squares methods can be viewed as modifications of the more familiar logistic regression and loglinear model methods, so the discussion will begin with a review of these in the simple case with only sampling error. Extensions to the case of sampling error plus a local overdispersion term will then be discussed, followed by a discussion of the rationale for modeling block effects as fixed rather than random. The focus will be on the binomial case.

5.1 Logistic regression

Consider the usual logistic regression, where $r \sim \mathrm{Bin}(n,p)$ with p determined by the fixed effect model

$$(5.1) \qquad \lambda(p) = \log(p/q) = x'\beta.$$

Here x' is a row vector describing the treatments applied, and β is a column vector of fixed parameters. The inverse logit can be written as

$$p(\lambda) = [1 + \exp(-\lambda)]^{-1}.$$

Expanding about $x'\beta$ we get the first order Taylor's approximation

$$p(\lambda) \approx p^o + \frac{dp}{d\lambda}(\lambda - \lambda^o)$$
$$\approx p^o + p^o q^o x'(\beta - \beta^o),$$

so

$$(5.2) \qquad r/n - p^o \approx p^o q^o x'(\beta - \beta^o) + e$$

where $E(e) = 0$, and $\mathrm{Var}(e) = p^o q^o/n$. Dividing both sides of (5.2) by the standard deviation of e gives

$$(5.3) \qquad z = \frac{r/n - p^{o}}{\sqrt{p^{o}q^{o}/n}} \approx \sqrt{np^{o}q^{o}}\, x'(\beta - \beta^{o}) + \epsilon$$

where $E(\epsilon) = 0$ and $Var(\epsilon) = 1$. In matrix notation

$$(5.4) \qquad E(z) \approx A(\beta - \beta^{o}); \quad Cov(z) \approx I.$$

where the notation supresses the dependence of z and A on β^{o}. Define $\delta = (\beta - \beta^{o})$. Then iterative least squares can be described as algorithm 1:

1. Given an initial β^{o}, calculate z and A;
2. Find the least squares solution δ, to $A'A\delta = A'z$;
3. If δ is small, stop;

   Otherwise add δ to β^{o},recalculate z and A, and go to 2.

This algorithm gives the binomial maximum likelihood estimates of β upon convergence. To see this, note that upon convergence, $\beta - \beta^{o} = 0$, so the normal equations, $A'(z - A(\beta - \beta^{o})) = 0$ become $A'z = \Sigma\, n(\hat{p} - p)x = 0$. The joint likelihood is $\prod \binom{n}{r} p^{r} q^{n-r}$. The log likelihood can be written $\mathcal{L} = \Sigma\, n[\hat{p} \cdot \log(p/q) + \log(q)]$, Taking derivatives with respect to each β_k gives the maximum likelihood equations : $\partial\mathcal{L}/\partial\beta_k = \Sigma\, n(\hat{p} - p)x_k = 0$, which are the same as the normal equations at convergence.

The computations are sometimes reorganized by rearranging equation (5.2) to give

$$(5.5) \qquad y = \lambda^{o} + \frac{1}{p^{o}q^{o}}\left(\frac{r}{n} - p^{o}\right) = x\beta + \epsilon/(np^{o}q^{o})^{1/2}$$

so $E(y) \approx x'\beta$ and $\mathrm{Var}(y) \approx \frac{1}{np^o q^o}$. Then y can be interpreted as a linear approximation to the link function, evaluated at $\hat{p}$. In matrix notation

(5.6) $$E(y) \approx X\beta \quad \text{and} \quad \mathrm{Cov}(y) \approx \mathrm{Diag}(np^{\circ}q^{\circ})^{-1}.$$

In this case we need to fit a weighted regression of y on x using weights $np^{\circ}q^{\circ}$, to get coefficients $\hat{\beta}$. The original X is used, and a new $\hat{\beta}$ is obtained at each step directly. This version of the algorithm can be described as

algorithm 2:

1. Get initial values of y and $V = \mathrm{Diag}(np^{\circ}q^{\circ})$;
2. Solve $X'VX\hat{\beta} = X'Vy$;
3. If $\hat{\beta}$ is close to its previous value, stop; otherwise re-calculate y and V, and go to 2.

The estimate of $\mathrm{Cov}(\hat{\beta})$ is $(X'VX)^{-1}$, where V is calculated at the final values of $p^{\circ} = p(x'\hat{\beta})$. Nelder and Wedderburn (1972) show that each step of this version of the algorithm is equivalent to a step in the Gauss-Newton method solution of the maximum likelihood equations. The data can be used to supply initial values in a straightforward manner.

5.2 Overdispersion

The methods described above are based on binomial data with means determined by fixed effects only. One way of incorporating overdispersion into the model is to add a random term to the linear predictor, replacing equation (5.1) with

$$(5.7) \qquad \lambda(p) = x'\beta + u$$

where the u terms for different observations are independent with mean 0 and variance σ^2. Equation (5.7) could serve as a model for a randomized block experiment if block effects were fixed and incorporated into β. This is not the view of randomized block experiments taken here, but treating the block effects as if they were fixed serves as a technical device to obtain an analysis. This will be discussed in more detail below. For now, think of the block effects as fixed parameters, incorporated into β with the treatment effects, and let u be the local overdispersion term.

After incorporating a random term in the linear predictor, equation (5.2) is replaced by

$$(5.8) \qquad r/n - p^o \approx p^o q^o x'(\beta - \beta^o) + p^o q^o u + e$$

so

$$\mathrm{Var}(r/n) \approx p^{o2} q^{o2} \sigma^2 + p^o q^o/n.$$

This differs from equation (2.7) only in neglecting the term $(1 - 1/n)$ that arises from the non-linearity of $E[\mathrm{Var}(r|p)]$. Dividing by $(np^o q^o)^{1/2}$ no longer results in unit variance so algorithm 1 loses some of its appeal. Equation (5.5) is replaced by

$$(5.9) \qquad y \approx \lambda^o + \frac{r/n - p^o}{p^o q^o} \approx x'\beta + u + \epsilon/(np^o q^o)^{1/2}.$$

In matrix notation we have

$$(5.10) \qquad E(y) \approx X\beta; \quad \text{and} \quad \mathrm{Cov}(y) \approx \sigma^2 I + V^{-1}$$

where $V^{-1} = \mathrm{Diag}(1/np^o q^o)$. Equations (5.9) and (5.10) express a function of the counts as the sum of a fixed term, a random overdispersion error and an error due to the binomial distribution.

If σ^2 were known, algorithm 2 could be modified by replacing the matrix of weights, V, with $\Sigma = (\sigma^2 I + V^{-1})^{-1}$. Call this modification algorithm 3. The estimated covariance of $\hat{\beta}$ is then $(X'\Sigma X)^{-1}$ where Σ is evaluated at $V(\hat{\beta})$. The problem with this procedure is of course that σ^2 is rarely if ever known.

5.3 Empirically weighted least squares

A suggestion due to Finney (1971) for dealing with extra-binomial variation is to estimate the regression coefficients, i.e. the fixed effects, by the usual binomial regression (a probit regression in his case) and inflate $\text{Cov}(\hat{\beta})$ by an amount estimated from the data. The inflation factor suggested is χ^2/d where $\chi^2 = \Sigma \frac{(r/n - \hat{p})^2}{\hat{p}\hat{q}/n}$, and d is the number of degrees of freedom for the regression. The motivation for this suggestion seems to be an assumption that the extra-binomial variance is proportional to the binomial.

If algorithm 1 is used, $\chi^2 = \hat{f}'\hat{f}$, where $\hat{f}$ is the final residual vector, i.e. z evaluated at the final estimate of β. So Finney's approach yields the estimator $\hat{\text{Cov}}(\hat{\beta}) = (\hat{f}'\hat{f}/d)(A'A)^{-1}$. This is the appropriate estimator if $\text{Cov}(\hat{f}) = \eta^2 I$ for some η^2.

As another way of looking at Finney's suggestion suppose $\mathrm{Var}[\sin^{-1}(2p - 1)] = \sigma^2$, a constant, i.e. the extra-binomial variance is homogeneous on the angular scale. Then we have

$$\mathrm{Var}(\hat{p}) \approx \pi(1 - \pi)[1/n + \sigma^2(1 - 1/n)].$$

If the denominators, n, are equal, this is proportional to $\pi(1 - \pi)$ regardless of σ^2. In this case $\mathrm{Cov}(\hat{f})$ is approximately of the form $\eta^2 I$. Algorithm 2 could be used for obtaining the estimate of β since the correct weight matrix is αV where $\alpha = 1 + (n - 1)\sigma^2$ and the constant α can be factored out of the normal equations of step 2. The covariance of $\hat{\beta}$ would then be $\alpha(X'VX)^{-1}$. If the n's differ substantially the appropriate weights for the regression depend on σ^2. If σ^2 were known, then the variance of $\hat{p}$ would be a known function of the mean and iterative least squares could be used, replacing the normal equations in algorithm 2 by $X'WVX\hat{\beta} = X'WVy$, where $W = \mathrm{Diag}[1 + (n - 1)\sigma^2]^{-1}$.

Returning to the model described by equation (5.7) in which $\mathrm{Var}[\log(p/q)] = \sigma^2$, a constant, we have

$$\mathrm{Var}(\hat{p}) \approx \pi(1 - \pi)/n + \pi^2(1 - \pi)^2(n - 1)\sigma^2.$$

Equal denominators do not help in this case, i.e. they do not necessarily make Σ proportional to V.

An appealing approach in the absence on an *a-priori* estimate of σ^2 is the following empirical weighting scheme.

EWLS:

1. Assume $\sigma^2 = 0$ and fit the fixed effects by the usual logistic regression (e.g. algorithm 2). If the lack of fit is explainable by the binomial distribution then stop; otherwise go to step 2.

2. Estimate σ^2 by setting the Pearson chi-square statistic equal to its expectation.

3. Fit by iterative weighted least squares assuming σ^2 is known to have the value estimated at step 2 (e.g. by algorithm 3). If the Pearson chi-squared statistic is close to its degrees of freedom, stop; otherwise go to step 2.

Since the empirical estimate of the overdispersion variance appears in the matrix of weights, the covariance of $\hat{\beta}$ is estimated as if σ^2 were known to be $\hat{\sigma}^2$, e.g. by $(X'\hat{\Sigma}X)^{-1}$ if the overdispersion is assumed constant on the logit scale, and by $(X'\hat{W}VX)^{-1}$ if the overdispersion is assumed constant on the angular scale, where $\hat{\Sigma}$ and $\hat{W}$ denote Σ and W evaluated at $\sigma^2 = \hat{\sigma}^2$. Variations on the above procedure have been described in detail by Williams(1982) in the binomial setting, and by Breslow(1984) in the Poisson setting. Both of these authors give instructions for fitting the models using the computer program GLIM. Pierce and Sands (1975) have examined the use of Finney's approach (i.e. the method based on the assumption that extra-binomial variation is homogeneous on the angular scale) when in fact the extraneous error is homogeneous on the logit scale. Note that for equal denominators, Finney's suggestion is a special case of the EWLS procedure that stops at step 2.

For the general case in which denominators may be unequal, Williams sets $E(\chi^2) = \Sigma_i\ w_i(1 - w_i v_i q_i)[1 + \sigma^2(n_i - 1)]$ when the variance is assumed constant on the angular scale, and $E(\chi^2) = \Sigma_i\ w_i(1 - w_i v_i q_i)[1 + \sigma^2(1 - 1/n_i)v_i]$ when the variance is assumed constant on the logit scale, where w_i is the ith diagonal element of $\hat{W}$, v_i is the ith diagonal element of $\hat{V}$, and q_i is the ith diagonal element of $X(X'\hat{W}\hat{V}X)^{-1}X'$, which is the estimated variance matrix of the linear predictor, $\hat{\lambda}$. The overdispersion variance, σ^2, can be estimated by substituting the observed Pearson chi-square statistic for the expectation and solving.

5.4 Block effects

In a randomized block experiment there are fixed treatment effects, random block effects, and a random residual error that is composed of an overdispersion effect and a sampling error. The EWLS approach described above can be applied if the block effects are treated as fixed, leaving the overdispersion and sampling error as random effects. The block effects introduce the only dependence between observations, so treating the block effects as fixed converts the model to one with independent observations that can be fit by the methods of the previous section. Although the block effects are random, treating them as if they were fixed appears to be a useful technical device for obtaining an analysis.

In the standard model for a balanced randomized block design, block effects subtract out of treatment contrasts. As a consequence, treating the block effects as fixed has no effect on the final inferences about treatment contrasts, provided that the block-by-treatment interaction (if present) is left as a random effect. The block variance affects the variance of treatment mean estimates, but not the variance of contrast estimates. It is the experimental error estimated by the residual mean square (assuming only one observation per cell) that determines the variance of a contrast. In the classical setting, the randomness of the block effects doesn't effect the choice of efficient estimators either. The best linear unbiased estimators (BLUEs) under a model with fixed block effects continue to be BLUEs under a model with random block effects.

In the case of counted responses, the block effects can add to the overdispersion if the link is misspecified, but the overdispersion is estimated and incorporated into the weights in the EWLS approach. The equivalence of BLUEs under fixed and random block effect models does not extend to enumerative responses, although in the absence of treatment effects, the distribution of the estimators that are BLUEs under the fixed block model do not depend on the block variance when used with the random block model.

If residual errors are small compared to block effects, the realized block effects will be well estimated. The binomial variances then depend mainly on the realized block effects, so it seems reasonable to base the estimate of binomial error on an estimated block effect. In other words, with large block effects it seems sensible to base the estimates of mean-dependent variance components on the means predicted from a model with both block and treatment effects rather than on a model with only treatment effects. The EWLS approach is mainly a method for dealing with variance heterogeneity. Since the differences between the variances of different counts arise at a very local level, rather strong conditioning is required to identify those differences.

It would be possible to devise an EWLS method that treated block effects as random, but it would present several formidable problems in addition to one mentioned above. One is that the quality of the delta method approximation to the unconditional variance of $\hat{p}$ will not be as good as for the approximation to $\mathrm{Var}(\hat{p}|b_j)$. Another problem is finding an estimator of the block variance. Kleinman (1975, p.739) estimates $\mathrm{Cov}(p_{1j},p_{2j})$ by $\Sigma_{j=1}^{J}(\hat{p}_{1j} - \bar{p}_{1\cdot})(\hat{p}_{2j} - \bar{p}_{2\cdot})/J$ (where $\bar{p}_{i\cdot}$is the mean of $\{\hat{p}_{i1},\ldots,\hat{p}_{iJ}\}$) and remarks that "no appropriate weighting scheme suggests itself." The obvious way of extending this approach to the case of more than two treatments is to estimate a separate covariance for each pair of treatments. But in order to model the dependence between counts for the same block by a single shared block effect, some sort of smoothing of the covariance matrix is required.

5.5 Analysis of caterpillar example

The caterpillar data was analyzed by five methods and the results summarized in Table 9. Two variations of the EWLS approach were tried, one based on overdispersion with constant variance on the angular scale, i.e. Finney's approach, and the other based on overdispersion that is constant on the logit scale, i.e based on the basic model of equations (2.1) and (2.2). The ANOVA approach was tried using both the angular transformation and the logit transformation. In addition, ordinary logistic regression based on purely binomial residuals was tried.

Table 9. Analysis of the caterpillar data

Method	trt	est.	se	t	p
logistic regression	1	-2.85	.55	-5.1	.003
(i.e binomial m.l.)	2	-3.00	.51	-5.9	.002
Deviance = 12.75, d.f. = 4					
emp. wt. l. s.	1	-2.85	.84	-3.4	.014
const. var. on ang. scale	2	-3.00	.78	-3.8	.009
est. of Var(e) = .07					
emp. wt. l. s.	1	-2.89	.88	-3.3	.015
const. var. on logit scale	2	-3.00	.85	-3.5	.012
est. ofVar(e) =.7					
ANOVA	1	-3.03	.85	-3.6	.012
logit transformation	2	-3.16	.85	-3.7	.010
ANOVA	1	-1.29	.37	-3.5	.013
angular transformation	2	-1.42	.37	-3.8	.009

In this example there is clear evidence of a treatment effect for both treatments, but no evidence of a difference in effect between the two treatments. All the methods are in general agreement except for the binomial maximum likelihood method, which, by ignoring the overdispersion, greatly overstates the significance.

The ANOVA after an angular transformation estimates the treatment effects on a different scale than the other methods, which estimate differences of logits. However the t-statistics for that method are in line with the other methods with the exception of the binomial maximum likelihood method.

5.6 Analyses of eelworm example

The eelworm example was also analyzed by a variety of methods, but the collection of methods was not entirely analogous to the collection used for the caterpillar data. Analyses of variance were done using counts, square roots of counts, and $\log(\hat{p})$. In the EWLS approach, the assumption that extra-Poisson error is constant on the square root scale was not tried. Instead, the extra-Poisson error was assumed to be constant on the logarithmic scale which would correspond to a multiplicative error, and weighted least squares was used with both counts and rates as the response variables. Breslow (1984) gives the GLIM macros that were used for fitting these two variations.

In the analysis based on rates, the response variable is $y = \log(r/n)$, so $\mathrm{Var}(y) \approx 1/(n\pi) + \sigma^2$. The algorithm starts by assuming $\sigma^2 = 0$, so y is fit by a weighted regression using r as weights, since r estimates $n\pi$. This is Poisson weighting, but because the calculations were done using GLIM with the normal error option, the standard error given in table 10 is based on the observed lack of fit, not on the binomial variance as a function of the mean. After obtaining estimates of the regression parameters, σ^2 is estimated by setting $\Sigma \frac{(y - x\hat{\beta})^2}{\hat{\sigma}^2 + 1/r}$ equal to the degrees of freedom, d. This equation is solved iteratively using

$$\hat{\sigma}^2 = (1/d)\ \Sigma \frac{(y - x\hat{\beta})^2}{1 + (r\hat{\sigma}o^2)^{-1}}$$

where $\hat{\sigma}^2$ is the updated estimate based on $\hat{\sigma}o^2$. The estimate, $\hat{\sigma}^2$ is then used to compute weights $w = n\hat{\pi} + 1/\hat{\sigma}^2$ which are used in the next fit. Iteration between fitting fixed effects and estimating σ^2 continue until the estimate of σ^2 stops changing appreciably.

The procedure based on counts rather than rates is similar, except that the Poisson error option in GLIM is used, so when Poisson weighting is used, the standard errors for the treatment effects are based on the estimated means rather than on the lack of fit. In other words, the analysis of counts starts with the Poisson maximum likelihood analysis. The resulting estimates, standard errors, and t and p values are reported in Table 10.

Table 10. Analysis of the eelworm data

Method	param	est	s.e.	t	p
ANOVA	dose	22.4	43.3	.52	.617
response = r	trt	26.1	43.3	.66	.560
ANOVA	dose	.408	1.59	.26	.802
response = √r	trt	1.44	1.59	.91	.386
ANOVA	dose	-.03	.275	-.10	.92
response = log(r/n)	trt	.32	.275	.15	.28
weighted l.s.	dose	.13	.19	.69	.50
response = log(r/n)	trt	.05	.20	.26	.80
Poisson weighting (wts = r)					
empirically wt. l.s.	dose	.01	.27	.05	.96
response = log(r/n)	trt	.30	.27	1.12	.29
est Var(log(p)) = .29					
Poisson maximum likelihood	dose	.09	.03	2.94	.015
response = log(r)	trt	.11	.03	3.44	.006
deviance = 412, df = 10					
empirically wt. l.s.	dose	.085	.25	.35	.73
response = log(r)	trt	.26	.25	1.06	.31

All methods except the Poisson maximum likelihood strongly suggest that any apparent treatment effect for either fumigant or dose could easily be due to chance. The Poisson maximum likelihood method, which is based on the assumption of no overdispersion, greatly overstates the significance of the effects. Little can be learned from comparing estimates, even for methods using the same scale, since the standard errors are so much larger than the estimates. It is mildly interesting to note that, using the log(r/n) response, the empirically weighted estimates lie between the estimates for the unweighted, and Poisson weighted methods, as might be expected.

6. DIRECT MAXIMUM LIKELIHOOD

6.1 The likelihood function

If we take model described by equations (2.1) and (2.2) and the accompanying conditions, and add the condition that b_j and u_{ij} are normally distributed, the likelihood can be written as:

$$L(\beta,\sigma_b^2,\sigma_u^2;r,n) = \prod_j \int \phi(b_j;\sigma_b^2) \prod_i \int \phi(u_{ij};\sigma_u^2)\, L_{ij}(p_{ij})\, du_{ij}\, db_j$$

where $\phi(a;\sigma^2)$ is the normal density of a with mean zero and variance σ^2;

where $p_{ij} = p(x_i\beta + b_j + u_{ij})$;

and where $L_{ij}(p) = \binom{n}{p}p^r(1-p)^{n-r}$, with $n = n_{ij}$, and $r = r_{ij}$.

6.2 Numerical integration

The likelihood can be evaluated numerically by repeated quadratures. The computer program that was written for this purpose used Gaussian quadrature, integrating with respect to the Gaussian density functions that occur in the likelihood. This method limited the program to relatively modest binomial denominators. When the binomial denominators are large, the relevant region in the domain of the binomial density is quite small, so either a very fine resolution is required, or the quadrature routine must search for the important domain of integration and focus sampling on the important regions. A second program that attempted such a search was written, but it was not running in time to be used.

6.3 Maximization

The program that was used maximized the likelihood using a Davidon-Fletcher-Powell (DFP) algorithm as described in Reklaitis, Ravindran, and Ragsdell(1983, p114). The DFP algorithm starts out as a gradient ascent method, but using the change in the function and its gradient to estimate a second derivative matrix, it gradually shifts to a Gauss-Newton method as the maximum is approached. The DFP algorithm does not require the computation of second derivatives, an important savings since the nested quadratures required much computation. However, the second derivatives of the loglikelihood were computed directly by nested quadratures once an approximate maximum was found. The reported standard errors for the estimated treatment effects are based on the negative inverse of the second derivative that was computed directly, and not on the estimated second derivative from the DFP algorithm.

The maximization required some interaction with the program. The general procedure was to start with the parameter values used to generate the data and let the program search for the maximum likelihood using 10 quadrature points for each integral. The program would stop when two successive iterations failed to change the parameter estimates by a given percentage, initially set at five percent, or when the program could not increase the likelihood in the direction of the gradient. The program was re-run using more quadrature points and starting with the parameter values that gave the low resolution maximum likelihood. Sixteen or nineteen quadrature points were used in the final run, depending on how much the parameter values had been changing at the end of the previous run. Changes in parameter values near the endpoint were generally less than five percent, and often less than one percent, with the larger percent changes generally being for estimates near zero. Increasing the number of quadrature points from sixteen to nineteen had very little effect on the treatment effect estimates or their standard errors.

7. SIMULATIONS AND COMPARISONS

7.1 Simulations

The direct maximum likelihood method is used here as a reference method, against which the other methods can be compared. The comparisons were done on data generated according to the logit-normal model, so the likelihood being maximized is known to be correct. No attempt at a Monte Carlo analysis was made. Interest centered on the behavior of different methods on individual data sets, rather than on summaries of their behavior over many trials.

The simulation results are summarized in Figures 1 - 4. In all cases, the experiment simulated consisted of a two-by-two factorial design in each of five blocks. The treatments had additive effects on the logit scale. Treatment one had no effect, while treatment two had the large effect of -2.2, which would reduce a response rate of .5 to a rate of .1. While these fixed effects remained the same in all trials, four combinations of block and local overdispersion variances were used with each variance assigned the value of 0 or .5 in a two-by-two factorial arrangement. For each of the four variance combinations, eight data sets or trials were generated.

Uniform random numbers were generated using the standard random number generator in GLIM and converted to standard normal random deviates by applying the inverse of the standard normal cumulative distribution function. These standard normal deviates were then multiplied by the appropriate standard deviation to provide deviates with the required variance. These were used as the random block and overdispersion effects, being added to the fixed effects to generate the logits of binomial probabilities. For each probability, 40 independent Bernoulli trials were simulated, and the number of successes recorded. The same random number seeds were used to generate corresponding trials for each of the four variance combinations, so the response probabilities in the four situations are based on the same standard normal deviates. This means that trial 1 with Var(b) = .5 and Var(u) = 0 will have the same realized block effects as trial 1 with Var(b) = .5 and Var(u) = .5.

7.2 Results

Six methods were used to estimate the treatment effects in each trial. Figures 1-4 give the estimated effects plus or minus two standard errors as estimated from a single trial. The methods are from left to right:

1. binomial maximum likelihood;
2. EWLS assuming overdispersion constant on the angular scale;
3. EWLS assuming overdispersion constant on the logit scale;
4. ANOVA using the logit transformation;

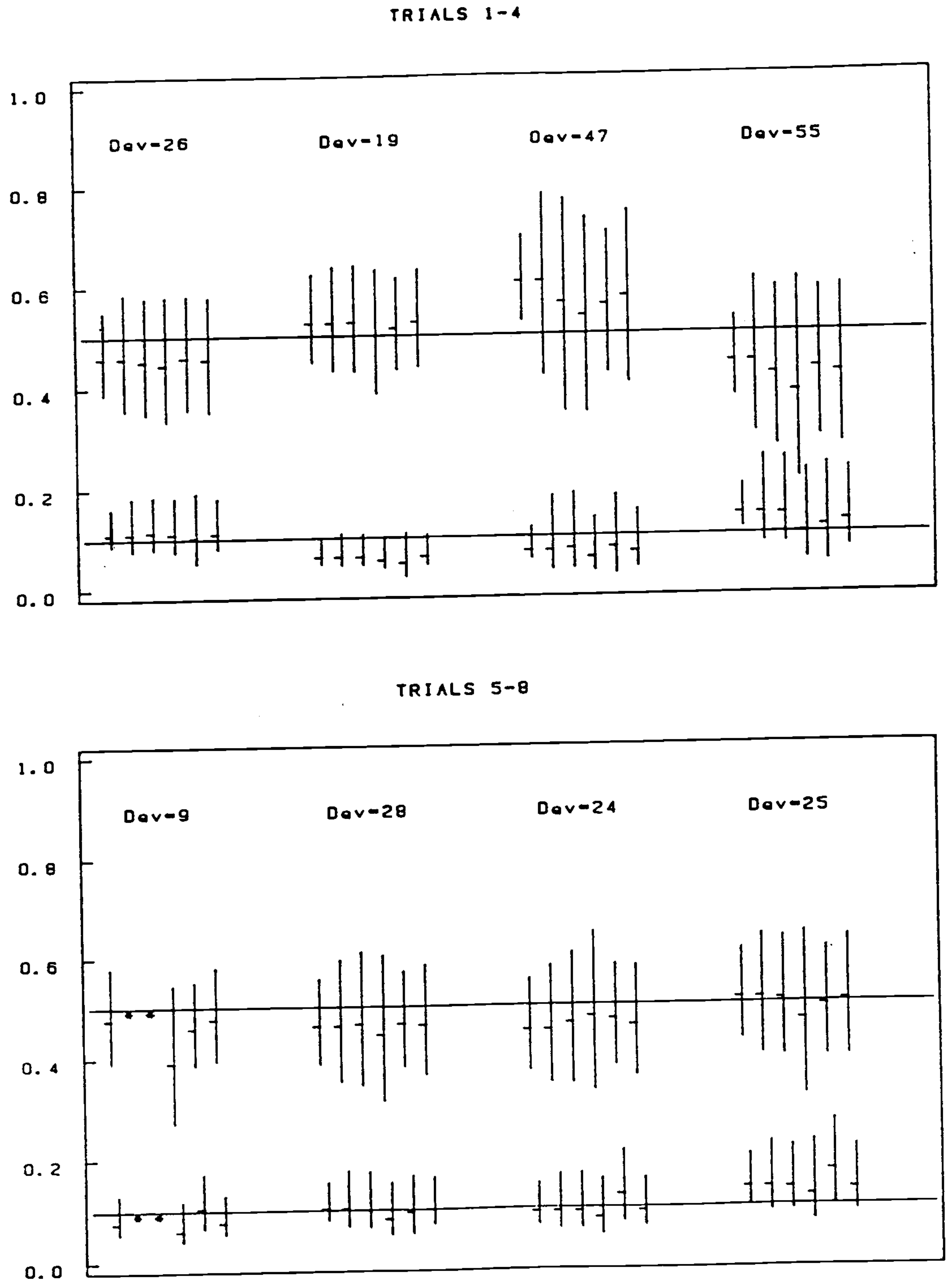
5. ANOVA using the angular transformation;

6. direct maximum likelihood as described in chapter 6.

The estimates were on the logit scale for all methods except the angular ANOVA. To facilitate comparisons, the estimates were converted to the response probabilities that would result from applying the estimated effect to a reference probability of .5. So the zero effect of treatment one on the logit scale translates to .5 on the probability scale, and the effect of treatment two, which was -2.2 on the logit scale, translates to .1. Since the grand mean parameter was set to zero and the random effects have zero means, the response probabilities corresponding to the expected values for the logit model are in fact .5 or .1 depending on the level of treatment two.

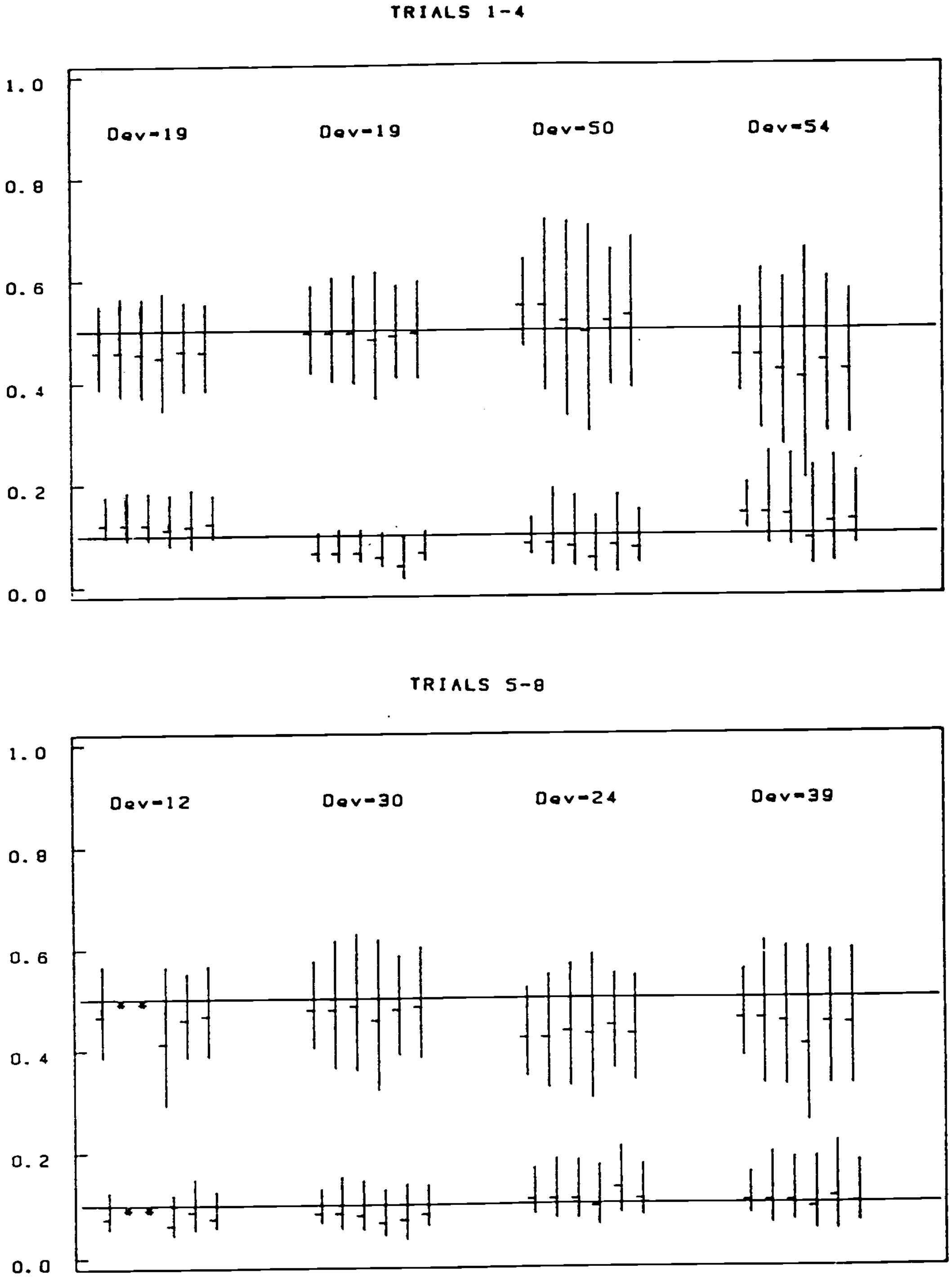
In figures 1 - 4 the estimates and intervals for the six methods applied to each trial are grouped together with the deviance from the binomial maximum likelihood fit written above. The horizontal lines show the probabilities corresponding to the true parameter values used for generating the data, .5 for the treatment one effect and .1 for the treatment two effect. Since the EWLS approach only prescribes going beyond a binomial maximum likelihood fit if the deviance or the Pearson chi-square statistic is large, empirical weighting is not applicable when the deviance is small. When the deviance was less than 13, the degrees of freedom, the binomial maximum likelihood estimate is also the EWLS estimate.

Figure 1
Simulation results: estimates +/- 2se on probability scale.
Var(b) = 0.5 Var(u) = 0.5



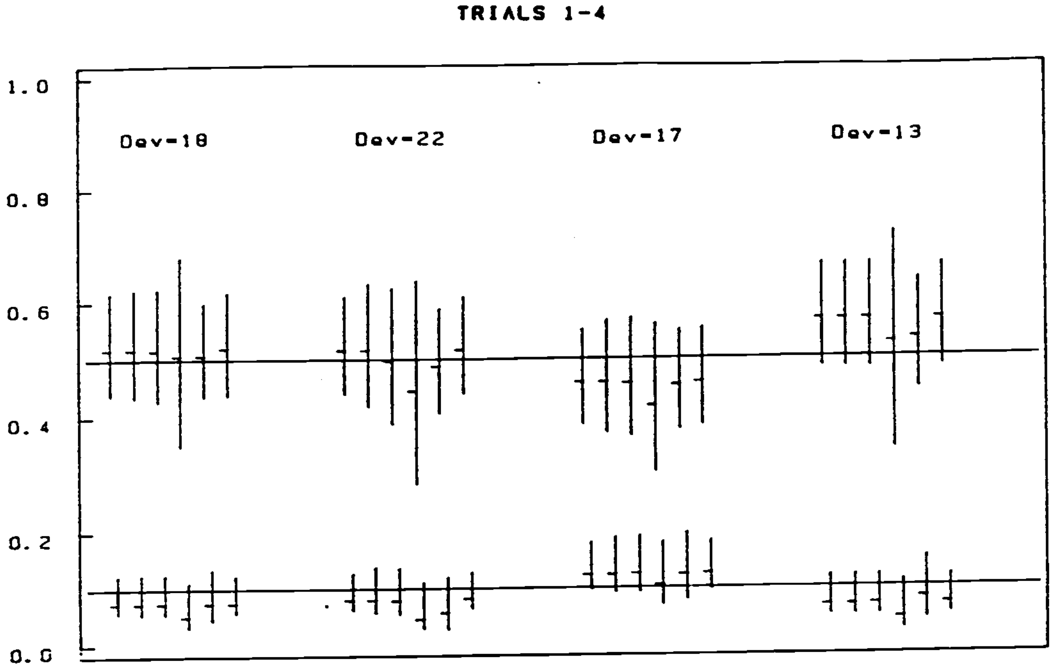
* NA - small deviance

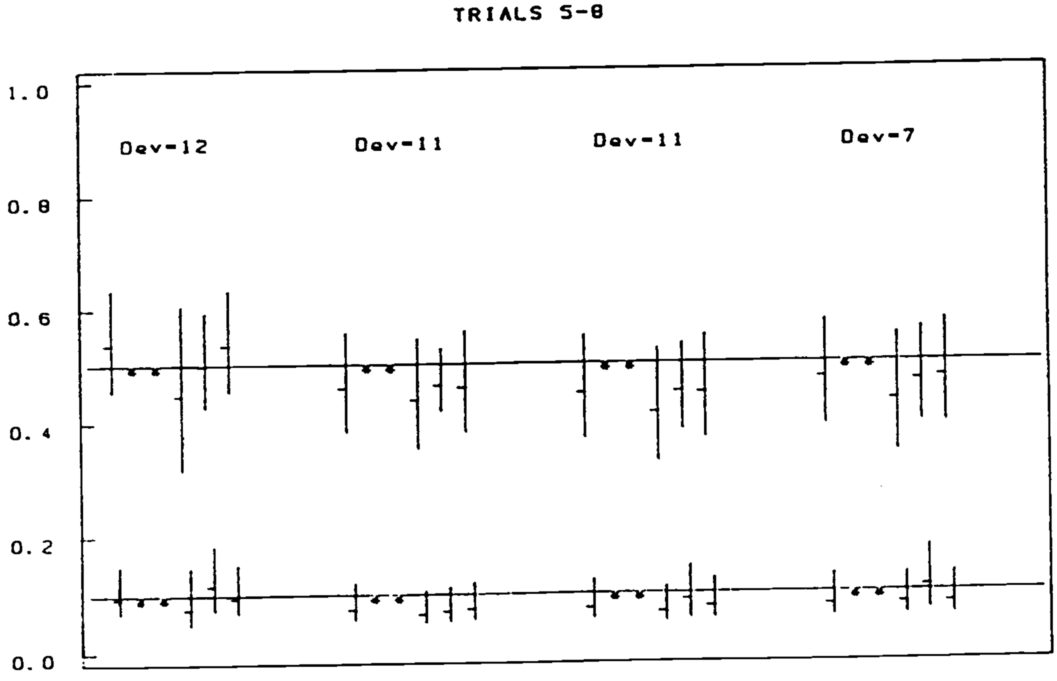
Figure 2
Simulation results: estimates +/- 2se on probability scale.
Var(b) = 0 Var(u) = 0.5



* NA - small deviance

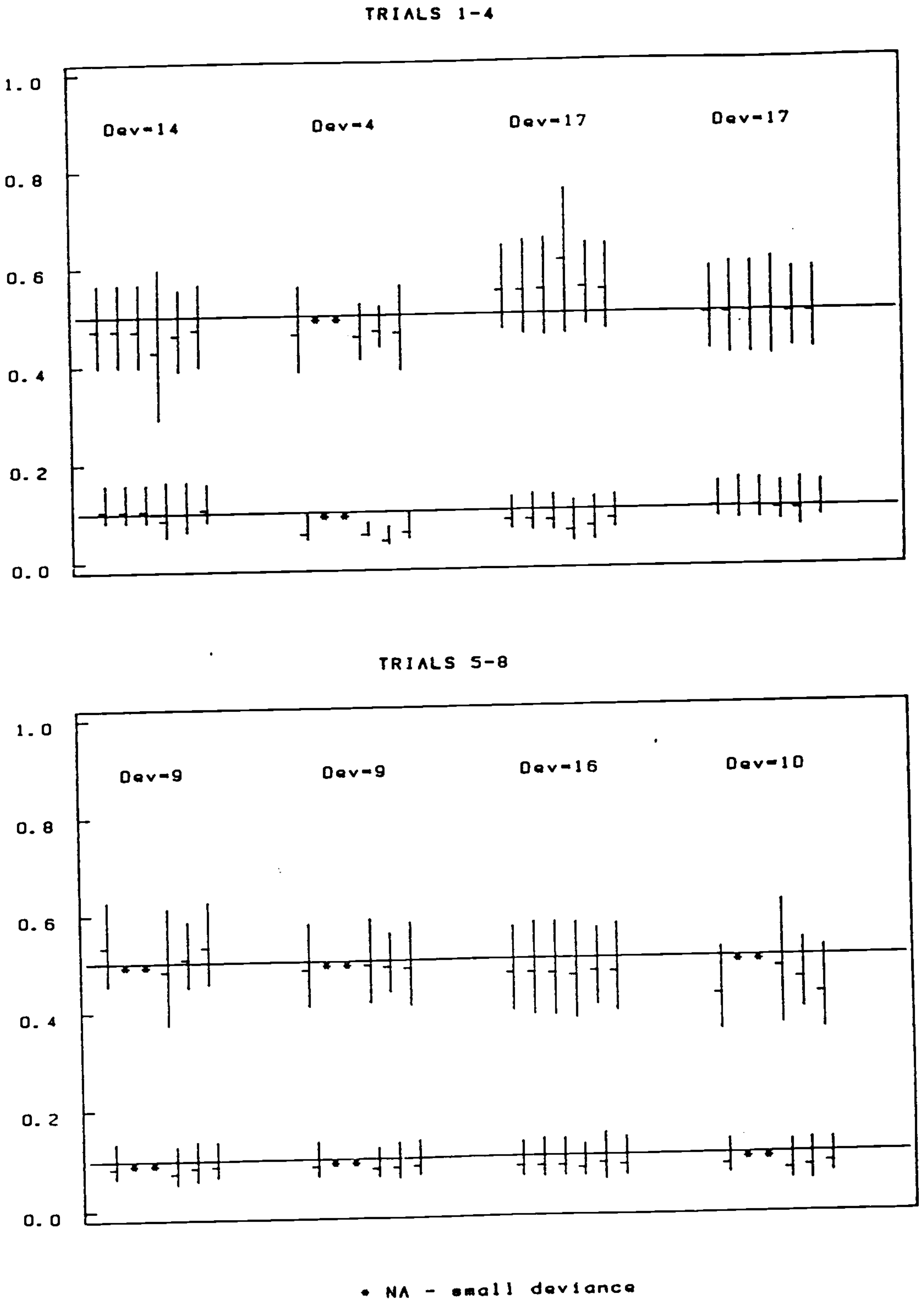
Figure 3
Simulation results: estimates +/- 2se on probability scale.
Var(b) = 0.5 Var(u) = 0





• NA - small deviance

Figure 4
Simulation results: estimates +/- 2se on probability scale.
Var(b) = 0   Var(u) = 0



7.3 Discussion of Simulation Results

One of the more striking results of these simulations is the range of deviances. In the case with both the block and local variances at .5, the deviance ranged from 9 to 55. The 90th percentile of the chi-square distribution with 13 degrees of freedom is approximately 20. The 95th percentile is near 22. So the samples ranged from no evidence of overdispersion to substantial evidence.

When the deviance was large, the intervals from the binomial maximum likelihood method were much shorter than those obtained by maximizing the correct likelihood. In the set of trials where both variances were large, the binomial maximum likelihood interval failed to cover the true value once for each parameter, and nearly missed .1 in two additional trials. For deviances less than 22 (the .05 level), the binomial m.l. method gave results quite consistent with the "correct" maximum likelihood method. There was no evidence of any problem with the policy of using the binomial model unless there is moderate evidence against it.

In the case with the smallest deviance, which was about four, both ANOVA estimates had short error bars and both failed to cover the effect of treatment two. The ANOVA approach uses only the lack of fit to estimate standard errors, while the other methods assume the binomial variance as a lower bound.

In general, the logit transformation tended to give reasonable estimates of the effect of treatment two but excessively long intervals for the effect of treatment one. The angular transformation sometimes resulted in short intervals for the treatment one effect, but it tended to give long intervals for the treatment two effect. Neither variation on the ANOVA approach tracked the maximum likelihood estimate as well as the weighted least squares methods.

The EWLS methods were both somewhat conservative compared to the direct maximum likelihood results when the deviance was large. When the deviance was moderate the EWLS, binomial maximum likelihood and complete maximum likelihood methods were very similar. When the deviance was small, the EWLS methods are of course identical with the binomial maximum likelihood method. The differences between the two variations on the EWLS approach were not striking but were noticable. The method based on constant variance on the logit scale tracked the maximum likelihood estimator more closely but gave more conservative intervals.

The occasional extreme conservatism of the ANOVA on the logit scale is hard to explain as it does not seem to depend on the amount of overdispersion or the observed deviance and is confined to the effect of treatment one. The angular ANOVA seemed better behaved, though still somewhat conservative for the larger effect, at least compared to the maximum likelihood method. With Var(u) = .5 and n = 40, the binomial error is still a substantial part of the residual error, so the logit transformation never really stabilizes the residual variance. If the binomial denominators were large, the angular ANOVA might have more difficulties, but in the situations considered one of the main faults of the ANOVA approach is the possibility of small intervals when the deviance is small.

The most evident features of the behavior of the three main approaches are that the binomial maximum likelihood method does not respond to evidence of overdispersion with expanded confidence intervals, that the EWLS approach responds to evidence of overdispersion but not to evidence of underdispersion, and that the ANOVA approach responds to both. Each approach may be appropriate depending on what mechanisms are thought to be at work in the experiment. In a typical randomized block experiment there is usually ample reason to expect overdispersion, but little if any reason to expect underdispersion, making EWLS the most appropriate approach in general.

The angular ANOVA seems to give reasonable results for the type of situation explored here, provided that some caution is used. If the overdispersion is small, some experiments may by chance fit an additive block and treatment effect model particularly well. The ANOVA method takes a small lack of fit at face value as an indication of small variance, ignoring the binomial sampling variance. In many situations the ANOVA method may have reasonable behavior over a large number of experiments, but in particular cases with small lack of fit statistics, the ANOVA results may be overly optimistic. If the overdispersion is substantial, such cases will not occur often. On the other hand, if the overdispersion is the dominant contribution to the residual error, as might be the case when denominators are large, the angular transformation might not do a very good job of stabilizing the variance.

The differences in performance between the two variations on the EWLS approach were not great, although the situation being simulated would be expected to favor the version based on constant variance on the logit scale. In a real situation the other method may perform relatively better than indicated here. When the binomial denominators are equal, Finney's suggestion is particularly simple. The savings in computation are trivial given reasonable computing power, but the conceptual simplicity of inflating the binomial variance to account for extra-binomial sources of error may be helpful in explaining the method. If the conceptual complexity of the EWLS approach poses a serious problem for a researcher, a compromise procedure might be to analyze the data initially by relying on the binomial distribution. The data might be treated as a multi-way contingency table with blocks as layers. A lack of fit statistic such as the Pearson chi-square could then be computed, and if it is unacceptably large, the angular ANOVA method could be used instead of empirical weighting. This sort of combined approach has no doubt been used often and solves the problem of overly optimistic error estimates from using ANOVA on data sets with a small lack of fit. The simulations presented here are of course too limited to allow a very general conclusion to be drawn, but they do seem to suggest that the EWLS approach would be preferable as an omnibus method, and that it gives a better approximation to a maximum likelihood analysis.

7.4 Summary

There are a number of aspects of any randomized block experiment that will be important in choosing a method for analysis; these include: the range of fixed effects, the block variance, the overdispersion variance, the size of denominators or of expected counts, the number of blocks, the scale on which the linear model is assumed, and the constancy and additivity of the random effects on that scale. In any randomized block experiment, the block variance can be expected to be substantial, and the overdispersion variance may often be substantial as well. In the two examples that were analyzed, the sampling distribution fit the data very poorly, and the analyses that were based on the sampling distribution grossly overstated the significance of the estimates.

If fixed effects are small, nonadditivity and variance heterogeneity (at least that due to the mean-variance relationship) are not serious problems. Unless block effects are also small, additivity and constant variance are still important to the decision as to how to combine the information from different blocks.

When the denominators are large, the sampling variance becomes a minor component in the total residual error. The choice of a transformation may then have to be made in light of the data, rather than relying on the angular of square root transformations. The simulations did not cover the large denominator situation, but they did include rather large amounts of overdispersion. Both the angular ANOVA and the two EWLS methods appeared to give reasonable results in general, with the ANOVA capable of giving overly optimistic standard errors if the data happened to fit an additive model especially well. When there was a large lack of fit, the EWLS methods tended to be conservative relative to the direct maximum likelihood method.

With a large number of blocks, standard errors could be estimated for contrasts without pooling variance estimators. With fewer blocks and a large main effect of secondary importance, partial pooling is a possible compromise. With still fewer blocks, the assumptions of additivity and constant variance are needed on the same scale for the ANOVA approach to be useful.

The EWLS approach allows additivity and constant variance to be assumed on different scales. In the binomial case with equal denominators, assuming constant variance on the angular scale leads to a particularly simple method, even when fixed effects are assumed additive on the logit scale. The simulations did not indicate a great deal of sensitivity to the choice between the logit or angular scale for assuming constant variance, even with rather large amounts of overdispersion. The sensitivity should be less with smaller amounts of overdispersion.

7.5 Suggestions for Future Work

The number of simulated data sets analyzed was quite modest, partly because the goal here was to get an initial impression of the various methods rather than to describe their behavior in detail and partly due to the difficulty in automating the direct maximum likelihood method. A more detailed simulation of a greater variety of situations would of course yield firmer conclusions.

The simulations presented here could be extended in several directions. The effects of large denominators and of unequal denominators remain to be seen The overdispersion variance of 0.5 was used because it seemed quite large. It was hoped that a large value would turn up any bad behavior that might be encountered in experiments with modest sample sizes, as well as indicate problems that might occur when sample sizes are larger. As mentioned above, the simulations did not cover the case where the binomial variance is only a minor component of the residual variation.

It would of course be interesting to do similar simulations for the Poisson situation. The strong analogy between the two situations would lead one to expect somewhat similar results for comparable combinations of Poisson and extra-Poisson variation.

Finally, the direct maximum likelihood method could no doubt be improved considerably. It was not seriously considered as a competitior to the other methods due to its computational complexity and the lack of available software. The maximum likelihood approach also involves the additional assumption of a particular distribution for the random effects, although it could prove reasonably robust in spite of this additional assumption. In short, the maximum likelihood approach appeared to be feasible but slow and hard to automate. For independent observations, Hinde (1982) has used GLIM to maximize a compound Poisson likelihood assuming a normally distributed error additive with the log link. If a similar simplification of the computations can be accomplished for the model with random block effects, the direct maximum likelihood approach may prove to be practical.

BIBLIOGRAPHY

Altham, P. E. (1971), "The Analysis of Matched Proportions," *Biometrika* 58, 561-576.

—— (1976), "Discrete Variable Analysis for Individuals Grouped into Families," *Biometrika*, 63, 263-269.

Anderson, D. A. and Aitkin, M. (1985), "Variance Component Models With Binary Response: Interviewer Variability," *Journal of the Royal Statistical Society*, Ser. B, 47, 203-210.

Anscombe (1954), Discussion of Fisher(1954). *Biometrics* 10, 141-144.

Baker, R. J. and Nelder, J. A. (1978) *The GLIM system, Release* 3, Oxford: Numerical Algorithms Group.

Barlow, R.E., Bartholomew, D.J., Bremner, J.M., Brunk, H.D. (1972) *Statistical Inference under Order Restrictions*. New York: John Wiley and Sons.

Bartlett, M.S. (1936) "The Square Root Transformation in Analysis of Variance" *Journal of the Royal Statistical Society*, Suplement, 68-78.

—— (1947) "The Use of Transformations," *Biometrics* 3, 39-52.

—— (1954), Discussion of Fisher (1954), *Biometrics* 10, 140-141.

Box, G.E.P. and Cox, D.R. (1964), "An Analysis of Transformations," *Journal of the Royal Statistical Society*, Ser. B, 26:211.

Breslow, N. E. (1984), "Extra-Poisson Variation in Log-linear Models," *Applied Statistics*, 33, 38-44.

Brier, S. S. (1980), "Analysis of Contingency Tables under Cluster Sampling," *Biometrika*, 67, 591-596.

Cochran, W. G. (1943), "Analysis of variances for percentages based on unequal numbers," *Journal of the American Statistical Association*, 38, 287-301.

—— (1947) "Some Consequences when the Assumptions for the Analysis of Variance Are Not Satisfied" *Biometrics* 3 pp22-38.

—— (1950), "The Comparison of Percentages in Matched Samples," *Biometrika*, 77, 256-266.

—— (1954) Discussion of Fisher(1954). *Biometrics* 10,144-147.

Cochran, W. G. and Cox, G (1957) *Experimental Design*. New York, John Wiley and Sons.

Cohen, J. E. (1976), "The Distribution of the Chi-Squared Statistic under Cluster Sampling from Contingency Tables," *Journal of the American Statistical Association*, 71, 665-670.

Daroch, J. N. (1981) "The Mantel-Haenszel Test of Marginal Symmetry; Fixed Effects and Mixed Models for a Categorical Response," *International Statistical Review*, 49, 285-307.

Draper, N.R. and Hunter, W.G. (1969), "Transformations: Some Examples Revisited," *Technometrics*, 11, 23-40.

Eisenhart, C. (1947) "The Assumptions Underlying the Analysis of Variance" *Biometrics* 3 pp1-22.

Finney, D. J. (1971), *Probit Analysis*, (third edition)London: Cambridge University Press.

Fisher, R. A. (1954) "The Analysis of Variance with Various Binomial Transformations," *Biometrics* 10,130-139.

Haseman, J. K. and Kupper, L. L. (1979), "Analysis of Dichotomous Response Data from Certain Toxocological Experiments, *Biometrics*, 35, 281-293.

Hinde, J. (1982), "Compound Poisson Regression Models," in *GLIM 82: Proceedings of the International Conference in Generalized Linear Models* (R. Gilcrest, ed.), pp. 109-121. Berlin: Springer-Verlag.

Hudson, H. M. (1983) Random Effects in Log-Linear Models," *Journal of Statistical Computing and Simulation*, 17, 43-50.

Kempthorne, J. (1982), The Analysis of Clustered Attribute Data from Nested and/or Classification Designs using Random Effects Models, University of North Carolina at Chapel Hill, Institute of Statistics Mimeo Series No. 1429.

Kleinman, J. C. (1973), "Proportions with Extraneous Variance: Single and Independent Samples," *Journal of the American Statistical Association*, 68, 46-54.

—— (1975), "Proportions with Extraneous Variance: Two Dependent Samples," *Biometrics* 31, 737-743.

Mantel, N. and Haenszel, W. (1959), "Statistical Aspects of the Analysis of Data from Retrospective Studies of Disease," *Journal of the National Cancer Institute*, 22, 719-748.

Nelder, J. A. and Wedderburn, R. W. M. (1972), "Generalized Linear Models," *Journal of the Royal Statistical Society*, Ser. A, 135, 370-384.

Pierce, D. A. (1976), "A Random Effects Model for Matched Pairs of Binomial Data," Technical Report 55, Oregon State University, Dept. of Statistics.

Pierce, D. A. and Sands B. R. (1975), "Extra-Bernoulli Variation in Binary Data," Technical Report 46, Oregon State University, Dept. of Statistics.

Reklaitis, Ravidran, and Ragsdell (1983), *Engineering Optimization, Methods and Applications*, New York: John Wiley and Sons.

Snedecor, G. W. and Cochran, W. G. (1980), *Statistical Methods* (seventh edition), Ames, Iowa: Iowa State University Press.

Tornqvist, Vartia, and Vartia (1985), "How Should Relative Changes be Measured?" *American Statistician*, 39, 43-46.

Tukey, J.W. (1949), "One Degree of Freedom for Non-Additivity" *Biometrics*, 5, 232-242.

Williams, D. A. (1982) "Extra-binomial variation in logistic linear models," *Applied Statistics*, 31, 144-148.

Zelen, M. (1981), "The Analysis of Several 2x2 Contingency Tables," *Biometrika*, 58, 129-137.

APPENDICES

APPENDIX 1

GLIM macros for fitting by EWLS

```
$SUB XBIN
$C
$C BASED ON WILLIAMS, APPLIED STATISTISCS 31(1982)144-148
$C
$PRI "  MACROS FOR EXTRA-BINOMIAL LOGISTIC REGRESSION"
$PRI ""
$PRI "  DECLARE COUNTS AS YVAR,"
$PRI "  BINOMIAL ERROR,"
$PRI "  SET WEIGHTS W = 1,"
$PRI "  FIT MODEL (FOR ANALYSIS BASED ON BINOMIAL DISTN)"
$PRI "  FOR EXTRA-BINOMIAL ERROR CONSTANT ON ANGULAR SCALE:"
$PRI "     USE ITA REPEATEDLY;"
$PRI "  FOR EXTRA-BINOMIAL ERROR CONSTANT ON LOGIT SCALE:"
$PRI "     USE ITL REPEATEDLY."
$PRI ""
$C
$C ----------------------------------------------------------------
$C
$MACRO IBWT!
$EXTRACT %VL $CA WVQ=%PW*%WT*%VL!
$CA %P=(%X2-%CU(%PW*(1-WVQ)))/(%CU((%BD-1)*%PW*(1-WVQ)))!
$CA W=1/(1+%P*(%BD-1))$!
$ENDMAC
$C
$MACRO ITA !
$USE IBWT !
$PRINT "ESTIMATED VAR. OF ARCSINE 2P-1 = "%P $!
$FIT . !
$PRINT "NEW CHI-SQUARE, D.F."%X2 %DF $ !
$ENDMAC
$C
$C -----------------------------------------------------------------
$C
$MACRO LHWT!
$EXTRACT %VL $CALC WVQ=%PW*%WT*%VL!
$CA %P=(%X2-%CU(%PW*(1-WVQ)))/%CU(%PW*(1-WVQ)*(1-1/%BD)*%WT)!
$CA W=1/(1+%P*(1-1/%BD)*%WT)!
$ENDMAC
$C
$MACRO ITL !
$USE LHWT !
$PRINT "ESTIMATED VAR. OF LOGIT P = "%P $!
$FIT . !
$PRINT "NEW CHI-SQUARE, D.F."%X2 %DF $ !
$ENDMAC
$C
$RETURN
```

```
$SUB XPSN
$C
$C BASED ON BRESLOW, APPLIED STATISTICS 33(1984)38-44
$C
$PRI "  MACROS FOR EXTRA-POISSON VARIATION"
$PRI ""
$PRI "  FOR EMPIRICAL ANALYSIS DECLARE LOG RATES AS YVAR,"
$PRI "  NORMAL ERROR,"
$PRI "  STORE 1 OVER NUMERATOR (COUNT) IN TAU2,"
$PRI "  INITIALIZE PRIOR WEIGHTS W TO NUMERATOR"
$PRI "  THEN FIT MODEL, CALL IN1 ONCE AND IT1 REPEATEDLY"
$PRI ""
$PRI "  FOR MAXIMUM-LIKELIHOOD ANALYSIS DECLARE NUMERATORS AS YVAR,"
$PRI "  POISSON ERROR,"
$PRI "  LOG DENOMINATORS AS OFFSET,"
$PRI "  AND INITIALIZE PRIOR WEIGHTS W TO 1"
$PRI "  THEN FIT MODEL, CALL IN2 ONCE, RECYCLE AND CALL IT2 REPEATEDLY"
$C
$MAC IN1 $EXT %VL $CAL WQ = %PW*(1-%PW*%VL/%SC) $!
$CAL %S=(%X2-%DF)/%CU(WQ) $!
$PRINT "INITIAL SIGMA2" $LOOK %S $ !
$ENDMAC
$C
$MAC SIG1 $CAL %R=%S !
$CAL %S=%CU((%YV-%FV)**2/(1+TAU2/%R))/%DF : %E=(%S-%R)/%S !
$CAL %E=%GT(%E*%E,.0000001) $ !
$ENDMAC
$C
$MAC IT1 $CAL %E=1 $WHILE %E SIG1 $CAL %G=%F !
$PRINT "NEXT SIGMA2" $LOOK %S $CAL W=1/(%S+TAU2) $FIT . !
$PRINT "NEW CHI-SQUARE, D.F."%X2 %DF $ !
$ENDMAC
$C
$MAC IN2 $EXT %VL $CAL WQ=%FV*(1-%FV*%VL) $!
$CAL %S=(%X2-%DF)/%CU(WQ) $!
$PRINT "INITIAL SIGMA2" $LOOK %S $ !
$ENDMAC
$C
$MAC SIG2 $CAL %R=%S $!
$CAL %S=%CU((%YV-%FV)**2/(%FV*(%FV+1/%R)))/%DF : %E=(%S-%R)/%S !
$CAL %E=%GT(%E*%E,.0000001) $ !
$ENDMAC
$C
$MAC IT2 $CAL %E=1 $WHILE %E SIG2 $CAL %G=%F !
$PRINT "NEXT SIGMA2" $LOOK %S $CAL W=1/(1+%S*%FV) $FIT . !
$PRINT "NEW CHI-SQUARE, D.F." %X2 %DF $ !
$ENDMAC
$C
$RETURN
```

APPENDIX 2

A Pascal program for computing maximum likelihood estimates

```
{ *************************  file = LIKE.PAS  **************************** }

PROGRAM LIKE;

  { MAXIMUM LIKELIHOOD ESTIMATION FOR A RANDOMIZED BLOCK EXPERIMENT }
  { WITH CONDITIONALLY BINOMIAL RESPONSES }

  { This program evaluates the likelihood by nested Gausian quadratures, }
  { and optionally searches for the parameter values that maximize the }
  { likelihood by means of the Davidon-Fletcher-Powell algorithm. }
  { The model assumed is that r[i,j] is binomial(n[i,j],p[i,j]), }
  { conditionally on p[i,j], where (ignoring some indexing details)}
  { logit(p[i,j]) = theta[i] + (block effect) + (local effect). }
  { Where the block and local effects are independent normal random deviates.}
  { Theta is a vector of mean logits, one for each treatment.  It is the  }
  { product of the design matrix Z and the parameter vector beta. }
  { NB:  For ease of maximization, theta, beta, and Z are all }
  { augmented to accomodate the logs of the block and local variances }
  { as the first two elements in the parameter vectors.  The values 1,2,... }
  { of the treatment index for the data correspond to the values T1,T2,... }
  { of the parameter index }

  { FILES:  }
  { 3 files are used.  The file LIKE.TBL, which contains the quadrature }
  { tables, must be available on the default drive.  The names of the output }
  { and data files are specified at the keyboard. }
  { Data is read freeform with the order of the }
  { columns being: block index, treatment index, count, denominator. }
  { The design matrix for the treatment effects, and the initial parameter }
  { values are entered from the keyboard. }

{ ------------------ CONSTANTS AND TYPES ------------------ }

CONST
  maxtrt = 4;        { NOTE:  If maxtrt is changed, Tmax, range of paramindex }
  maxblk = 10;       { and CASE statement in DFP must also be changed. }
  maxqpts = 20;      { max. no. of quadrature points }

TYPE
  paramindex = (B,E,T1,T2,T3,T4);
  DataType = ARRAY[1..maxtrt,1..maxblk] OF INTEGER;
  QTblType = ARRAY[1..30] OF REAL;    { for Quadrature Tables }
  probtype = ARRAY[1..maxtrt,1..maxqpts,1..maxqpts] OF REAL;
  swtype = 0..2;      { switch for calling loglike }


CONST
  Tmax = T4;

TYPE
  vector = ARRAY[B..Tmax] OF REAL;
  matrix = ARRAY[B..Tmax,B..Tmax] OF REAL;
  plabeltype = ARRAY[B..Tmax] OF STRING[15];

CONST
  paramlabel : plabeltype = ('Ln blk variance','Ln int variance',
                             'Beta 1         ','Beta 2         ',
                             'Beta 3         ','Beta 4         ');
```

```
{ ------------------------ VARIABLES -------------------------- }

VAR

            { DATA: }

  r,n : DataType;            { r = binom. count; n = denom. }
  ntrts, nblks : INTEGER;    { numbers of trts & blocks }
  lastb,
  lasttrt : paramindex;      { position of last trt param in vector }

            { PARAMETERS: }

  theta,                 { full parameter vector used for likelihood evaluation }
  beta,                  { parameter for maximization under a linear model }
  db,                    { incremental change (calculated) }
  se  : vector;          { std. error of beta }
  p  : probtype;         { prob. of response }
  Z, Ztr : matrix;       { augmented design matrix and its transpose }

            { LIKELIHOOD: }

  L : REAL;                       { log likelihood }
  dL, dLdbeta : vector;           { first derivative of log likelihood }
  ddL, ddLdbeta, H : matrix;      { second derivs, DFP metric }

            { QUADRATURE TABLE: }

  nqpts  : INTEGER;       { number of quadrature pts. }
  A, u,                   { A = weights, u = arguments }
  bu, eu : QTblType;      { change of variable fns of u }

            { MISC. }

  F, datafile : TEXT;
  flname, dflname : STRING[14];
  crit : REAL;
  maxcycles : INTEGER;
  response : CHAR;
  di, dj : paramindex;

{ ----------------------- INCLUDED FILES ----------------------- }

{$I UTIL.PAS} { functions: Power    procedures: GetParameters }
              {            MaxI                 GetData       }
              {            Norm                 WriteVec      }
              {            Mnorm                WriteMat      }
              {                                 WriteLike     }
              {                                 CnvrtVec      }
              {                                 CnvrtMat      }
              {                                 Piv           }
              {                                 NegInvert     }
              {                                 GetQtable     }

{$I LOGLIKE.PAS} { procedure:  LogLike }
                 {               CalcConstants }
                 {               CalcGs  }
                 {               CalcDs  }
                 {               CalcDDs }
```

```
{$I SEARCH.PAS} { procedures:   SetBounds }
                {               TightenBounds }
                {                 CheckInterior }
                {                 Shorten }
                {               Iset  }
                {               CalcH }
                {               DFP   }

{ -------------------- MAIN PROGRAM ---------------------- }

BEGIN { PROGRAM }

  { Open output file }

  WRITELN;
  WRITE('ENTER NAME OF OUTPUT FILE [default = screen]>  ');
  READLN(flname);
  IF LENGTH(flname) = 0
    THEN
      flname := 'CON:          ';
  ASSIGN(F,flname);
  REWRITE(F);

  { Get data and initial parameter values }

  WRITELN;
  WRITE('ENTER NAME OF DATA FILE:  ');
  READLN(dflname);
  ASSIGN(datafile,dflname);
  GetData(datafile,nblks,ntrts,r,n);
  WRITELN('Blocks:  ',nblks,'   Treatments:   ',ntrts);
  lasttrt := paramindex(ntrts + 1);
  GetParameters(ntrts,beta,Z,lastb);

  { Initialization chores }

  FOR di := B TO Tmax DO
    FOR dj := B TO Tmax DO
      Ztr[di,dj] := Z[dj,di];    { Ztr = transpose of Z }
  WRITELN(F,'DATAFILE:   ',dflname);
  WRITELN(F,'Blocks:   ',nblks,'   Treatments:   ',ntrts);
  WRITELN(F,'theta = treatment means (logit scale)');
  WRITELN(F,'beta  = (Z)(theta)');
  WRITELN(F,'Z:');
  FOR di := T1 TO lasttrt DO
    BEGIN
    FOR dj := T1 TO lastb DO
      WRITE(F,'   ',Z[di,dj]:8:4);
    WRITELN(F);
    END;

  { Get the quadrature resolution }

  WRITELN;
  WRITE('ENTER NUMBER OF QUADRATURE POINTS:  ');
  READLN(nqpts);
  WRITELN;
  WRITELN(F,'NO. QUADRATURE POINTS:  ',nqpts:3);
  GetQTable(nqpts,A,u);
  { Optionally, maximize likelihood }

  WRITE('Maximize Likelihood? (Y or N): ');
  READLN(response);
  IF (response = 'Y') OR (response = 'y')
    THEN
      BEGIN
      WRITELN;
      WRITELN('STOPPING CRITERIA');
      WRITE('  ENTER MAXIMUM PERCENT CHANGE IN ESTIMATES:  ');
      READLN(crit);
      crit := crit/100;
      FOR di := B TO lastb DO
        db[di] := crit * beta[di];
      WRITE('  ENTER MAXIMUM NUMBER OF CYCLES: ');
      READLN(maxcycles);
      WRITELN;
      WRITELN(F,'STOPPING CRITERIA:');
      WRITELN(F,'  Max change in beta: ',crit:1,'   Max cycles: ',maxcycles);
      DFP(beta,db, crit,maxcycles); { *** search for maximum *** }
      WRITELN(F);
      END; { IF yes }

  { Final evaluation and report }

  WRITELN(F);
  WRITELN(F);
  WRITELN(F,'---------- FINAL VALUES ----------');
  WRITELN(F);

  WRITELN('*** FINAL VALUES: ***');
  WriteVec(OUTPUT,beta,lastb);
  WRITELN('*** CALCULATING LOGLIKELIHOOD AND DERIVATIVES ***');

  CnvrtVec(theta, Z,beta);
  LogLike(2,theta,L,dL,ddL);
  CnvrtVec(dLdbeta, Ztr,dL);
  CnvrtMat(ddLdbeta, Z,ddL);
  WriteLike(F,L,dLdbeta,ddLdbeta,lastb);

  WRITELN('*** INVERTING HESSIAN ***');

  NegInvert(ddLdbeta, lastb);
  WRITELN(F);
  WRITELN(F,'NEG. INV. HESSIAN:');
  WriteMat(F,ddLdbeta,lastb);
  WRITELN(F);

  { Standard errors }

  FOR di := B TO lastb DO
    se[di] := SQRT(ddLdbeta[di,di]);

  WRITE(F,'BETA:     ');
  WriteVec(F,beta,lastb);
  WRITE(F,'S.E.:     ');
  WriteVec(F,se,lastb);

  WRITELN('***** DONE *****');
  CLOSE(F);
END.
```

```
{ ************************  FILE = UTIL.PAS  ************************* }

FUNCTION Power(base, exponent : REAL): REAL;
  BEGIN
    Power := EXP(exponent * LN(base));
  END; { Power }

{ ------------------------------------------------------------------- }

FUNCTION MaxI(a,b : INTEGER) : INTEGER;
  BEGIN
    IF a >= b
      THEN MaxI := a
      ELSE MaxI := b;
  END; { MaxI }

{ ------------------------------------------------------------------- }

FUNCTION Norm(vec : vector; last : paramindex) : REAL;
  VAR
    ss : REAL;
    pi : paramindex;
  BEGIN
    ss := 0;
    FOR pi := B TO last DO
      ss := ss + SQR(vec[pi]);
    Norm := SQRT(ss);
  END;

{ ------------------------------------------------------------------- }

FUNCTION Mnorm(change,reference : vector): REAL;
  VAR
    max, temp : REAL;
    pi : paramindex;
  BEGIN
    max := 0;
    FOR pi := B TO lastb DO
      BEGIN
      IF reference[pi] <> 0
        THEN
          temp := ABS(change[pi]/reference[pi])
        ELSE
          temp := 0;
      IF temp > max
        THEN max := temp;
      END;
    Mnorm := max;
  END; { Mnorm }

{ ------------------------------------------------------------------- }
```

```
PROCEDURE GetParameters(numtrts : INTEGER;
                        VAR beta : vector;
                        VAR Z : matrix;
                        VAR lastb : paramindex);
  VAR
    pi, pj : paramindex;
    blength, i,j : INTEGER;

  BEGIN
    FOR di := B TO Tmax DO     { initialize the parameter and model matrix }
      BEGIN
      beta[di] := 0;
      FOR dj := B TO Tmax DO
        Z[di,dj] := 0;
      END; { FOR di }
    Z[B,B] := 1;
    Z[E,E] := 1;


    WRITELN;
    WRITE('Enter length of parameter for treatment effect model: ');
    READLN(blength);
    lastb := paramindex(blength + 1);

    WRITELN;
    WRITELN('Enter model matrix:');
    WRITE('         ');
    FOR i := 1 TO blength DO
      WRITE(i:2);
    WRITELN;
    pi := E;
    FOR i := 1 TO numtrts DO
      BEGIN
      pi := SUCC(pi);
      WRITE('ROW',i:2,':  ');
      CASE lastb OF
        T1 : READLN(Z[pi,T1]);
        T2 : READLN(Z[pi,T1],Z[pi,T2]);
        T3 : READLN(Z[pi,T1],Z[pi,T2],Z[pi,T3]);
        T4 : READLN(Z[pi,T1],Z[pi,T2],Z[pi,T3],Z[pi,T4]);
      END; {CASE}
      WRITELN;
      END; { FOR i }

    WRITELN;
    WRITELN('ENTER INITIAL PARAMETER VALUES:');
    WRITELN('          param       value');
    FOR pi := B to lastb DO
      BEGIN
      WRITE(paramlabel[pi],':  ');
      READLN(beta[pi]);
      END;

   END; { GetParameters }
 { ---------------------------------------------------------------- }
```

```
PROCEDURE GetData(VAR datafile : TEXT;
                  VAR numblks,numtrts : INTEGER;
                  VAR r,n : DataType);

  VAR
    dflname : STRING[16];              { name of data file }
    t, b : INTEGER;                    { trt and block indices }
    rtemp, ntemp   : INTEGER;

  BEGIN
    FOR t := 1 TO maxtrt DO
      FOR b := 1 TO maxblk DO
        BEGIN
        n[t,b] := 0;
        r[t,b] := -1;
        END; {FOR}
    RESET(datafile);
    numtrts := 0;
    numblks := 0;
    REPEAT
      READLN(datafile,b,t,rtemp,ntemp);
      r[t,b] := rtemp;
      n[t,b] := ntemp;
      numtrts := MaxI(numtrts,t);
      numblks := MaxI(numblks,b);
    UNTIL(EOF(datafile));
  END; { GetData }

{ ---------------------------------------------------------------------- }

PROCEDURE WriteVec(VAR F : TEXT; vec : vector; last : paramindex);
  VAR
    di : paramindex;
  BEGIN
    FOR di := B TO last DO
      WRITE(F,'   ',vec[di]:9:5);
    WRITELN(F);
  END;

{ ---------------------------------------------------------------------- }

PROCEDURE WriteMat(VAR F : TEXT; M : matrix; last : paramindex);
  VAR
    di, dj : paramindex;
  BEGIN
    FOR di := B TO last DO
      BEGIN
      FOR dj := B TO last DO
        WRITE(F,M[di,dj]:9:5,'   ');
      WRITELN(F);
      END; { FOR di }
  END; { WriteMatrix }

{ ---------------------------------------------------------------------- }
```

```
PROCEDURE WriteLike(VAR F : TEXT;         { WRITES LIKELIHOOD AND DERIVATIVES }
                    L : REAL;
                    dL : vector;
                    ddL : matrix;
                    last : paramindex);
  BEGIN
    WRITELN(F);
    WRITELN(F,'LOG LIKELIHOOD = ',L);
    WRITELN(F);
    WRITELN(F,'DERIVATIVE OF LOG LIKELIHOOD:');
    WRITELN(F);
    FOR di := B TO last DO
      WRITE(F,dL[di]:9:5,'   ');
    WRITELN(F);
    WRITELN(F);
    WRITELN(F,'SECOND DERIVATIVES OF LOG LIKELIHOOD:');
    WriteMat(F,ddL,last);
    WRITELN(F);
  END; { WriteLike }

{ ----------------------------------------------------------------------- }

PROCEDURE CnvrtVec(VAR theta : vector; Z : matrix; beta : vector);
  VAR
    di, dj : paramindex;
  BEGIN
    FOR di := B TO Tmax DO
      BEGIN
      theta[di] := 0;
      FOR dj := B TO Tmax DO
        theta[di] := theta[di] + Z[di,dj] * beta[dj];
      END; { FOR di }
  END; { Convert }

{ ----------------------------------------------------------------------- }

PROCEDURE CnvrtMat(VAR H2 : matrix; Z, H1 : matrix);
  VAR
    di, dj, dk : paramindex;
    vec1, vec2 : vector;
  BEGIN
    FOR dj := B TO Tmax DO
      BEGIN
      FOR di := B TO Tmax DO
        vec1[di] := Z[di,dj];
      CnvrtVec(vec2, H1,vec1);
      FOR di := B TO Tmax DO
        BEGIN
        H2[di,dj] := 0;
        FOR dk := B TO Tmax DO
          H2[di,dj] := H2[di,dj] + Z[dk,di] * vec2[dk];
        END; { FOR di }
      END; { FOR dj }
  END; { Convert }

{ ----------------------------------------------------------------------- }
```

```
PROCEDURE Piv(VAR S : matrix; dk, last : paramindex);

  CONST
    tolerance = 0.000001;

  VAR
    pivot, magnitude : REAL;
    di,dj : paramindex;
    k : INTEGER;

  BEGIN { Piv }

    pivot := S[dk,dk];

    magnitude := SQRT(SQR(pivot));
    IF magnitude < tolerance
      THEN
        BEGIN
        k := ORD(dk);
        WRITELN('TOLERANCE ERROR AT PIVOT ',k:2);
        END
      ELSE
        BEGIN

          FOR di := B TO last DO
            IF di <> dk THEN
              FOR dj := B TO last DO
                IF dj <> dk THEN
                  S[di,dj] := S[di,dj] - S[di,dk] * S[dk,dj] / pivot;

          FOR di := B TO last DO
            IF di <> dk
              THEN
                BEGIN
                S[di,dk] := S[di,dk] / pivot;
                S[dk,di] := S[dk,di] / pivot;
                END; { IF }

          S[dk,dk] := -1 / pivot;

        END; { ELSE }

  END; { Piv }
 { ------------------------------------------------------------------- }

 PROCEDURE NegInvert(VAR hess : matrix; last : paramindex);

 VAR
   dk : paramindex;

 BEGIN { NegInvert }
   FOR dk := B TO last DO
     piv(hess,dk,last);
 END; { NegInvert }

 { ------------------------------------------------------------------- }
```

```
PROCEDURE GetQTable(nqpts : INTEGER; VAR a,u : QTblType);

{ Finds and reads Gaussian Quadrature table for the desired number of points }

CONST
  filename = 'LIKE.TBL';
VAR
  Qfile : TEXT;                      { source file for Gaussian Quad. tbl. }
  tblno,                             { no. points for the tbl. being read }
  index, j : INTEGER;
  utemp,atemp : REAL;
BEGIN
  ASSIGN(Qfile,filename);            { Open the table file }
  RESET(Qfile);
  REPEAT                             { Find and read appropriate table }
    READLN(Qfile,tblno);
    FOR index := 1 TO 15 DO
      BEGIN
      READLN(Qfile,utemp,atemp);
      u[index] := utemp;
      a[index] := atemp;
      END;
  UNTIL (tblno = nqpts);
  FOR index := 1 TO TRUNC((nqpts + 1)/2) DO    { Expand to symmetric table }
    BEGIN
      j := nqpts - index + 1;
      u[j] := -u[index];
      a[j] := a[index];
    END;
  CLOSE(Qfile);
END; { PROCEDURE GetQTable }
```

```
{ ************************ FILE = LOGLIKE.PAS ************************ }

PROCEDURE LogLike(switch : swtype;      { highest order derivative desired }
                  theta : vector;       { parameter }
                  VAR L : REAL;         { Log likelihood }
                  VAR dL : vector;      { first derivative of L wrt theta }
                  VAR ddL : matrix);    { second derivative }

  VAR
    G4,       { conditional binomial likelihood }
    dG4dT,
    ddG4dT2 : ARRAY[1..maxtrt,1..maxqpts,1..maxqpts] OF REAL;

    G3,       { Likelihood for one obsnervation given b }
    dG3dE,
    dG3dT,
    ddG3dT2,
    ddG3dET,
    ddG3dEE : ARRAY[1..maxtrt,1..maxqpts] OF REAL;

    G2,       { Conditional (given b) likelihood for one block }
    dG2dE,
    ddG2dEE : ARRAY[1..maxqpts] OF REAL;
    dG2dT,
    ddG2dET : ARRAY[1..maxtrt,1..maxqpts] OF REAL;
    ddG2dTT : ARRAY[1..maxtrt,1..maxtrt,1..maxqpts] OF REAL;

    G1,       { Likelihood for one block }
    dG1dB,
    dG1dE,
    ddG1dEE,
    ddG1dBB,
    ddG1dBE : REAL;
    dG1dT,
    ddG1dET,
    ddG1dBT : ARRAY[1..maxtrt] OF REAL;
    ddG1dTT : ARRAY[1..maxtrt,1..maxtrt] OF REAL;

    G : REAL;  { Likelihood }

    j, i, ii : INTEGER;
    di, dj : paramindex;

{ --------------------------- LOCAL PROCEDURES ------------------------- }

PROCEDURE CalcConstants;

  VAR
    i,m,mb,me : INTEGER;
    lambda : REAL;
    di : paramindex;

  BEGIN { CalcConstants }

    { calculate bu and eu: }

    FOR m := 1 TO nqpts DO
      BEGIN
      bu[m] := u[m] * SQRT(2 * EXP(theta[B]));
      eu[m] := u[m] * SQRT(2 * EXP(theta[E]));
      END; { FOR }
    { calculate p: }

    di := E;
    FOR i := 1 TO ntrts DO
      BEGIN
      di := SUCC(di);
      FOR mb := 1 TO nqpts DO
        FOR me := 1 TO nqpts DO
          BEGIN
          lambda := theta[di] + bu[mb] + eu[me];
          p[i,mb,me] := 1 / (1 + EXP(-lambda));
          IF (p[i,mb,me] = 1.0) OR (p[i,mb,me] = 0.0)
            THEN
              WRITELN('P = ',p[i,mb,me],' ;  indices:   ',i:4,mb:4,me:4);
          END; { FOR }
      END; { FOR i }

  END; { CalcConstants }


{ ---------------------------------------------------------------------- }


PROCEDURE CalcGs(j : INTEGER);

  VAR
    i,mb,me : INTEGER;
    sum, prod : REAL;
    pp, q, successes, failures : REAL;

  BEGIN

  { calculate G4: }

  FOR i := 1 TO ntrts DO
    FOR mb := 1 TO nqpts DO
      FOR me := 1 TO nqpts DO
        IF n[i,j] > 0
          THEN
            BEGIN
            pp := p[i,mb,me];
            q := 1 - pp;
            successes := r[i,j];                { Force type changes }
            failures := n[i,j] - r[i,j];
            G4[i,mb,me] := Power(pp,successes) * Power(q,failures);
            END
          ELSE
            G4[i,mb,me] := 0;

  { calculate G3: }

  FOR i := 1 TO ntrts DO
    FOR mb := 1 TO nqpts DO
      IF n[i,j] > 0
        THEN
          BEGIN
          sum := 0;
          FOR me := 1 TO nqpts DO
            sum := sum + A[me] * G4[i,mb,me];
          G3[i,mb] := sum;
          END
        ELSE
          G3[i,mb] := 1;
  { calculate G2 }

  FOR mb := 1 TO nqpts DO
    BEGIN
    prod := 1;
    FOR i := 1 TO ntrts DO
      IF n[i,j] > 0
        THEN
          prod := prod * G3[i,mb];
    G2[mb] := prod;
    END; { FOR }

  { calculate G1 }

  sum := 0;
  FOR mb := 1 TO nqpts DO
    sum := sum + A[mb] * G2[mb];
  G1 := sum;

  END; { CalcGs }

{ --------------------------------------------------------------------- }


PROCEDURE CalcDs(j : INTEGER);

VAR
  i,s,mb,me : INTEGER;
  pp,q,succ,fail,sum : REAL;

BEGIN

  { -- DERIVATIVES WRT TREATMENT EFFECTS -- }

  FOR s := 1 TO ntrts DO
    IF n[s,j] > 0
      THEN
        BEGIN

        { dG4dT }

        FOR mb := 1 TO nqpts DO
          FOR me := 1 TO nqpts DO
            BEGIN
            pp := p[s,mb,me];
            q := 1 - pp;
            succ := r[s,j];
            fail := n[s,j] - succ;
            dG4dT[s,mb,me] := succ * Power(pp,succ) * Power(q,(fail + 1))
                              - fail * Power(pp,(succ + 1)) * Power(q,fail);
            END; { FOR me }

        { dG3dT }

        FOR mb := 1 TO nqpts DO
          BEGIN
          sum := 0.0;
          FOR me := 1 TO nqpts DO
            sum := sum + A[me] * dG4dT[s,mb,me];
          dG3dT[s,mb] := sum;
          END; { FOR mb }
          { dG2dT }

          FOR mb := 1 TO nqpts DO
            dG2dT[s,mb] := dG3dT[s,mb] * G2[mb] / G3[s,mb];

          { dG1dT }

        sum := 0.0;
        FOR mb := 1 TO nqpts DO
          sum := sum + A[mb] * dG2dT[s,mb];
        dG1dT[s] := sum;

        END; { IF }

  { -- DERIVATIVES WRT LOG VARIANCE OF INTERACTION -- }

  { dG3dE }

  FOR mb := 1 TO nqpts DO
    FOR i := 1 TO ntrts DO
      IF n[i,j] > 0
        THEN
          BEGIN
          sum := 0.0;
          FOR me := 1 TO nqpts DO
            sum := sum + A[me] * (SQR(u[me]) - 0.5) * G4[i,mb,me];
          dG3dE[i,mb] := sum;
          END
        ELSE
          dG3dE[i,mb] := 0;

  { dG2dE }

  FOR mb := 1 TO nqpts DO
    BEGIN
    sum := 0.0;
    FOR i := 1 TO ntrts DO
      IF n[i,j] > 0
        THEN
          sum := sum + dG3dE[i,mb] * G2[mb] / G3[i,mb];
    dG2dE[mb] := sum;
    END; { FOR mb }

  { dG1dE }

  sum := 0.0;
  FOR mb := 1 TO nqpts DO
    sum := sum + A[mb] * dG2dE[mb];
  dG1dE := sum;


  { -- DERIVATIVES WRT VARIANCE OF LOG BLOCK EFFECT -- }

  sum := 0.0;
  FOR mb := 1 TO nqpts DO
    sum := sum + A[mb] * (SQR(u[mb]) - 0.5) * G2[mb];
  dG1dB := sum;

END; { CalcDs }
```

```
{ ---------------------------------------------------------------------- }

PROCEDURE CalcDDs(j : INTEGER);
VAR
  sum, pp, q, succ, fail, coeff, subsum : REAL;
  mb, me, s, t, i, k : INTEGER;

BEGIN

  { -- Derivatives wrt Ts and Tt -- }

  FOR s := 1 TO ntrts DO
    IF n[s,j] > 0
     THEN
      BEGIN

      { ddG4dT2 }

      FOR mb := 1 TO nqpts DO
       FOR me := 1 to nqpts DO
        BEGIN
        pp := P[s,mb,me];
        q := 1 - pp;
        succ := r[s,j];
        fail := n[s,j] - succ;
        coeff := 2 * SQR(succ) - 2 * succ * n[s,j] - n[s,j];
        ddG4dT2[s,mb,me] := SQR(succ) * Power(pp,succ)     * Power(q,fail + 2)
                          + coeff     * Power(pp,succ + 1) * Power(q,fail + 1)
                          + SQR(fail) * Power(pp,succ + 2) * power(q,fail);
        END; { FOR me }

      { ddG3dT2 }

      FOR mb := 1 TO nqpts DO
        BEGIN
        sum := 0;
        FOR me := 1 TO nqpts DO
          sum := sum + A[me] * ddG4dT2[s,mb,me];
        ddG3dT2[s,mb] := sum;
        END; { FOR mb }

      FOR t := 1 TO ntrts DO
         IF n[t,j] > 0
           THEN
             BEGIN

             { ddG2dTT }

             FOR mb := 1 TO nqpts DO
               BEGIN

               IF t = s
                 THEN
                   ddG2dTT[s,s,mb] := ddG3dT2[s,mb] * G2[mb] / G3[s,mb];

               IF t > s
                 THEN
                   ddG2dTT[s,t,mb] := dG3dT[s,mb] * dG3dT[t,mb]
                                    * G2[mb] / G3[s,mb] / G3[t,mb];
```

```
            IF t < s
              THEN
                ddG2dTT[s,t,mb] := ddG2dTT[t,s,mb];

            END; { FOR mb }

          { ddG1dTT }

          sum := 0;
          FOR mb := 1 TO nqpts DO
            sum := sum + A[mb] * ddG2dTT[s,t,mb];
          ddG1dTT[s,t] := sum;

          END
        ELSE
          ddG1dTT[s,t] := 0;

    END
   ELSE
    FOR t := 1 TO ntrts DO
      ddG1dTT[s,t] := 0;


{ -- Derivatives wrt Ts and E -- }

FOR s := 1 TO ntrts DO
  IF n[s,j] > 0
    THEN
      BEGIN

      { ddG3dET }

      FOR mb := 1 TO nqpts DO
        BEGIN
        sum := 0;
        FOR me := 1 TO nqpts DO
         sum := sum + A[me] * (SQR(u[me]) - 0.5) * dG4dT[s,mb,me];
        ddG3dET[s,mb] := sum;
        END; { FOR mb }

      { ddG2dET }

      FOR mb := 1 TO nqpts DO
        BEGIN
        sum := 0;
        FOR i := 1 TO ntrts DO
          IF (i <> s) AND (n[i,j] > 0)
            THEN
              sum := sum + dG3dE[i,mb] * G2[mb] / G3[s,mb] / G3[i,mb];
        ddG2dET[s,mb] := ddG3dET[s,mb] * G2[mb] / G3[s,mb]
                         + dG3dT[s,mb] * sum;
        END; { FOR mb }

      { ddG1dET }

      sum := 0;
      FOR mb := 1 TO nqpts DO
        sum := sum + A[mb] * ddG2dET[s,mb];
      ddG1dET[s] := sum;
      END
    ELSE
      ddG1dET[s] := 0;

{ -- Derivatives wrt Ts and B -- }

{ ddG1dBT }

FOR s := 1 TO ntrts DO
  IF n[s,j] > 0
    THEN
      BEGIN
      sum := 0;
      FOR mb := 1 TO nqpts DO
        sum := sum + A[mb] * ( SQR(u[mb]) - 0.5 ) * dG2dT[s,mb];
      ddG1dBT[s] := sum;
      END
    ELSE
      ddG1dBT[s] := 0;

{ -- Second derivative wrt E -- }

{ ddG3dEE }

FOR mb := 1 TO nqpts DO
  FOR i := 1 TO ntrts DO
    IF n[i,j] > 0
      THEN
        BEGIN
        sum := 0;
        FOR me := 1 TO nqpts DO
          sum := sum + A[me] * ( SQR(SQR(u[me])) - 2 * SQR(u[me]) + 0.25 )
                 * G4[i,mb,me];
        ddG3dEE[i,mb] := sum;
        END; { IF }

{ ddG2dEE }

FOR mb := 1 TO nqpts DO
  BEGIN
  sum := 0;
  FOR i := 1 TO ntrts DO
    IF n[i,j] > 0
      THEN
        BEGIN
        subsum := 0;
        FOR k := 1 TO ntrts DO
          IF k <> i THEN
            subsum := subsum + dG3dE[k,mb] * G2[mb] / G3[i,mb] / G3[k,mb];
        sum := sum + ddG3dEE[i,mb] * G2[mb] / G3[i,mb] + dG3dE[i,mb] * subsum;
        END; { FOR i }
  ddG2dEE[mb] := sum;
  END; { FOR mb }

{ ddG1dEE }

sum := 0;
FOR mb := 1 TO nqpts DO
  sum := sum + A[mb] * ddG2dEE[mb];
ddG1dEE := sum;
```

```
  { -- Second derivatives wrt B -- }

  { ddG1dBB }

  sum := 0;
  FOR mb := 1 TO nqpts DO
    sum := sum + A[mb] * ( SQR(SQR(u[mb])) - 2 * SQR(u[mb]) + 0.25 ) * G2[mb];
  ddG1dBB := sum;

  { -- Second mixed partial wrt B & E -- }

  { ddG1dBE }

  sum := 0;
  FOR mb := 1 TO nqpts DO
    sum := sum + A[mb] * ( SQR(u[mb]) - 0.5 ) * dG2dE[mb];
  ddG1dBE := sum;

END; { CalcDDs }

{ ---------------------------------------------------------------------- }


  BEGIN { LogLike }

    { initializations }

    CalcConstants;

    G := 1.0;

    IF switch >= 1
      THEN
        FOR di := B TO Tmax DO
          dL[di] := 0;

    IF switch >= 2
      THEN
        FOR di := B TO Tmax DO
          FOR dj := B TO Tmax DO
            ddL[di,dj] := 0;

    { For each block, call procs to calculate likelihood and derivatives, }
    { and convert to derivatives of the log likelihood }

    FOR j := 1 TO nblks DO
      BEGIN

      CalcGs(j);                                       { Likelihood }
      G := G * G1;
    IF switch >= 1                                      { First Derivatives }
      THEN
        BEGIN
        CalcDs(j);
        dL[B] := dL[B] + dG1dB / G1;
        dL[E] := dL[E] + dG1dE / G1;
        di := E;
        FOR i := 1 TO ntrts DO
          BEGIN
          di := SUCC(di);
          dL[di] := dL[di] + dG1dT[i] / G1;
          END;
        WRITE('.');
        END; { IF }

    IF switch >= 2                                      { Second Derivatives }
      THEN
        BEGIN
        CalcDDs(j);
        ddL[B,B] := ddL[B,B] + (ddG1dBB / G1) - (SQR(dG1dB) / SQR(G1));
        ddL[E,E] := ddL[E,E] + (ddG1dEE / G1) - (SQR(dG1dE) / SQR(G1));
        ddL[B,E] := ddL[B,E] + (ddG1dBE / G1) - (dG1dB * dG1dE / SQR(G1));
        ddL[E,B] := ddL[B,E];
        di := E;
        FOR i := 1 TO ntrts DO
          BEGIN
          dj := di;
          di := SUCC(di);
          ddL[B,di] := ddL[B,di] + (ddG1dBT[i] / G1)
                       - (dG1dB * dG1dT[i] / SQR(G1));
          ddL[di,B] := ddL[B,di];
          ddL[E,di] := ddL[E,di] + (ddG1dET[i] / G1)
                       - (dG1dE * dG1dT[i] / SQR(G1));
          ddL[di,E] := ddL[E,di];
          FOR ii := i TO ntrts DO
            BEGIN
            dj := SUCC(dj);
            ddL[di,dj] := ddL[di,dj] + (ddG1dTT[i,ii] / G1)
                               - (dG1dT[i] * dG1dT[ii] / SQR(G1));
            ddL[dj,di] := ddL[di,dj];
            END; { FOR ii }
          END; { FOR i }
          WRITE('. ');
        END; { IF }
    END; { FOR }

  L := LN(G);       { log likelihood }

END; { LogLike }
```

```
{ ************************* FILE = SEARCH.PAS  ************************* }

PROCEDURE SetBounds (L,delta,crit : REAL;
                      beta, dir : vector;
                      VAR bound : vector;
                      VAR Lb : REAL;
                      VAR stuck : BOOLEAN );

  { Sets bounds for max likelihood search along direction dir from beta. }
  { Delta is step length, betalow is nearer beta, betahigh is further. }

  CONST
    stretch = 2;
  VAR
    Lold : REAL;
    theta, betaold : vector;
    i : paramindex;

  BEGIN
    FOR i := B TO lastb DO
      bound[i] := beta[i] + delta * dir[i];
    CnvrtVec(theta, Z,bound);
    LogLike(O,theta,Lb,dL,ddL);
    WRITELN('L(bound) = ',Lb:9:5);
    stuck := FALSE;

    IF Lb < L             { step back until Lb > L then use previous step }
      THEN
        BEGIN
        REPEAT
          delta := delta / stretch;
          betaold := bound;
          Lold := Lb;
          FOR i := B TO lastb DO
            bound[i] := beta[i] + delta * dir[i];
          CnvrtVec(theta, Z,bound);
          LogLike(O,theta,Lb,dL,ddL);
          WRITELN('Stepping back.  L(bound) = ',Lb:9:5);
          stuck := ABS(Lb - Lold) < (crit/100);
        UNTIL (Lb >= L) OR stuck;
        bound := betaold;
        Lb := Lold;
        END
      ELSE { Lb >= L }     { Step forward until Lb < L }
        REPEAT
          delta := delta * stretch;
          betaold := bound;
          Lold := Lb;
          FOR i := B TO lastb DO
            bound[i] := beta[i] + delta * dir[i];
          CnvrtVec(theta, Z,bound);
          LogLike(O,theta,Lb,dL,ddL);
          WRITELN('Stepping forward.  L(bound) = ',Lb:9:5);
        UNTIL Lb <= Lold;
   END; { SetBounds }

 { --------------------------------------------------------------------- }
```

```
PROCEDURE TightenBounds ( crit : REAL;
                          VAR betalow, betahigh : vector;
                          VAR Llow, Lhigh : REAL );

          { ** Golden Section Interval Refinement Algorithm ** }

  CONST
    maxsteps = 20;

  VAR
    t : REAL;        { Golden section proportion }
    LC, LT, ssd, width : REAL;
    nsteps : INTEGER;
    i : paramindex;
    betaT, betaC : vector;

  PROCEDURE CheckInterior;
    BEGIN
    ssd := 0;
    FOR i := B TO lastb DO
      ssd := ssd + SQR(betahigh[i] - betalow[i]);
    width := SQRT(ssd);
    FOR i := B TO lastb DO
      BEGIN
      betaT[i] := betalow[i] + t*(betahigh[i] - betalow[i]);
      betaC[i] := betalow[i] + t*t*(betahigh[i] - betalow[i]);
      END;
    CnvrtVec(theta, Z,betaT);
    LogLike(0,theta,LT,dL,ddL);
    CnvrtVec(theta, Z,betaC);
    LogLike(0,theta,LC,dL,ddL);
    END;

   PROCEDURE Shorten(VAR TL,TC,TT,TH : vector; VAR LL,LC,LT,LH,W : REAL);
     BEGIN
       FOR i := B TO lastb DO
         BEGIN
         TH[i] := TT[i];
         TT[i] := TC[i];
         TC[i] := TL[i] + TH[i] - TT[i];
         END; { FOR }
       LH := LT;
       LT := LC;
       CnvrtVec(theta, Z,TC);
       LogLike(0,theta,LC,dL,ddL);
       ssd := 0;
       FOR i := B TO lastb DO
         ssd := ssd + SQR(TH[i] - TL[i]);
       W := SQRT(ssd);
     END; { Shorten }

   BEGIN { TightenBounds }

     t := (SQRT(5) - 1)/2;
     CheckInterior;
     nsteps := 0;
   REPEAT
     nsteps := nsteps + 1;
     WRITE(nsteps:3);
     IF LC > LT
       THEN Shorten(betalow,betaC,betaT,betahigh,Llow,LC,LT,Lhigh,width);
     IF LC < LT
       THEN Shorten(betahigh,betaT,betaC,betalow,Lhigh,LT,LC,Llow,width);
     IF LC = LT
       THEN
         BEGIN
         FOR i := B TO lastb DO
           BEGIN
           betahigh[i] := betaT[i];
           betalow[i]  := betaC[i];
           END; { FOR }
         Lhigh := LT;
         Llow  := LC;
         CheckInterior;
         WRITELN('*** EQUAL VALUES ***');
         END; { IF }
   UNTIL ( ABS(Lhigh - Llow) < crit ) OR ( nsteps > maxsteps );
   WRITELN;

  END; { TightenBounds }
{ ---------------------------------------------------------------------- }

PROCEDURE Iset(VAR H : matrix);  { Sets matrix to Identity }
  VAR
    di, dj : paramindex;
  BEGIN
    FOR di := B TO lastb DO
    FOR dj := B TO lastb DO
      IF (di = dj)
        THEN H[di,dj] := 1.0
        ELSE H[di,dj] := 0.0;
  END; {Iset}
{ ---------------------------------------------------------------------- }

PROCEDURE CalcH (H : matrix; dt, dg : vector; VAR NewH : matrix);

{ Updates the metric matrix of the DFP algorithm }

  VAR
    a1, a2, temp : REAL;
    i,j : paramindex;
    Hdg : vector;

  BEGIN { CalcH }
    a1 := 0;
    a2 := 0;
    FOR i := B TO lastb DO
      BEGIN
      a1 := a1 - dt[i] * dg[i];
      temp := 0;
      FOR j := B TO lastb DO
        temp := temp + H[i,j] * dg[j];
      a2 := a2 + dg[i] * temp;
      END; { FOR i }
    FOR i := B TO lastb DO
      BEGIN
      Hdg[i] := 0;
      FOR j := B TO lastb DO
        Hdg[i] := Hdg[i] + H[i,j] * dg[j] ;
      END; { FOR i }

    FOR i := B TO lastb DO
      FOR j := B TO lastb DO
        NewH[i,j] := H[i,j] + dt[i]*dt[j]/a1 - Hdg[i]*Hdg[j]/a2;

  END; { CalcH }

{ ---------------------------------------------------------------------- }

PROCEDURE DFP(VAR beta, db : vector; crit : REAL; maxcycles : INTEGER);

  VAR
    theta,
    oldbeta,
    bound,
    dir : vector;        { direction of search for new param value}

    L, L1, L2, deltaL : REAL;        { log likelihood }
    dL, dLdbeta,                     { first derivative of log likelihood }
    dLold, dgrad : vector;      { previous value and change of dL }
    ddL, H : matrix;                 { second deriv, DFP metric }
    di, dj : paramindex;             { array indices for derivatives }

    oldL, linecrit, delta : REAL;
    cycle : INTEGER;
    near, nearnow, nearlast,
    stuck, stucknow, stucklast : BOOLEAN;

  BEGIN { DFP }

  { ------- Calculate likelihood & derivatives for initial values ------ }

  WRITELN;
  WRITELN('*** CALCULATING LOGLIKELIHOOD AND GRADIENT ***');
  CnvrtVec(theta, Z,beta);
  LogLike(1,theta,L,dL,ddL);
  CnvrtVec(dLdbeta, Ztr,dL);
  WRITELN(F);
  WRITE(F,'INITIAL BETA: ');
  WriteVec(F,beta,lastb);
  WRITELN(F,'Log Likelihood = ',L:9:5);
  WRITE(F,'GRADIENT: ');
  WriteVec(F,dLdbeta,lastb);

  { ---------     Davidon - Fletcher - Powell Algorithm      --------- }

  Iset(H);              { initialize H to identity }
  nearlast := FALSE;
  stucklast := FALSE;
  cycle := 0;
  REPEAT
    IF NOT stucklast
      THEN cycle := cycle + 1;
```

```
WRITELN;
WRITELN('*** CYCLE:   ',cycle:2);
WRITELN;
WRITELN('DFP METRIC:');
WriteMat(OUTPUT,H,lastb);
WRITELN;

WRITELN(F);
WRITELN(F);
WRITELN(F,'CYCLE:   ',cycle:2);
WRITELN(F);

FOR di := B TO lastb DO           { H * dLdbeta = new direction }
  BEGIN
  dir[di] := 0;
  FOR dj := B TO lastb DO
    dir[di] := dir[di] + H[di,dj] * dLdbeta[dj];
  END;
WRITE('DIRECTION:');
WriteVec(OUTPUT,dir,lastb);

WRITELN('*** SETTING BOUNDS FOR SEARCH ***');

delta := 0.5 * Norm(db,lastb)/Norm(dir,lastb);
SetBounds(L,delta,crit,beta,dir, bound,L2,stucknow);
IF stucknow
  THEN
    BEGIN
    WRITELN('STUCK');
    WRITELN(F,'DFP ALGORITHM STUCK');
    IF stucknow AND NOT stucklast
      THEN
        BEGIN
        WRITELN('RESETTING DFP MATRIX');
        Iset(H)
        END;
    stuck := stucknow AND stucklast;
    END { IF }
  ELSE { NOT stucknow }
    BEGIN
    WRITELN('*** SEARCHING ALONG DIRECTION ***');
    oldbeta := beta;
    IF cycle = 1
      THEN
        linecrit := crit * 10
      ELSE
        linecrit := deltaL/100;
    TightenBounds(linecrit, beta,bound,L1,L2);
    FOR di := B TO lastb DO
      BEGIN
      beta[di] := (beta[di] + bound[di]) / 2;
      db[di] := beta[di] - oldbeta[di];
      END;

    nearnow := Mnorm(db,oldbeta) <= crit;
    near := nearnow AND nearlast;
    nearlast := nearnow;
```

```
      WRITE(F,'NEW BETA:    ');
      WriteVec(F,beta,lastb);
      WRITE(F,'CHANGE:      ');
      WriteVec(F,db,lastb);

      WRITE('NEW BETA:    ');
      WriteVec(OUTPUT,beta,lastb);
      WRITE('CHANGE:      ');
      WriteVec(OUTPUT,db,lastb);

      oldL := L;
      dLold := dLdbeta;

      { update H for DFP algorithm }

      WRITELN('*** CALCULATING LOGLIKELIHOOD AND GRADIENT ***');
      CnvrtVec(theta, Z,beta);
      LogLike(1,theta,L,dL,ddL);
      CnvrtVec(dLdbeta, Ztr,dL);

      FOR di := B TO lastb DO
        dgrad[di] := dLdbeta[di] - dLold[di];
      CalcH(H,db,dgrad,H);

      deltaL := L - oldL;
      WRITELN(F,'LOGLIKELIHOOD: ',L:9:5,'     CHANGE:   ',deltaL:8:5);
      WRITE(F,'GRADIENT:    ');
      WriteVec(F,dLdbeta,lastb);

      WRITELN('LOGLIKELIHOOD:   ',L:9:5,'     CHANGE:   ',deltaL:8:5);
      WRITE('GRADIENT:    ');
      WriteVec(OUTPUT,dLdbeta,lastb);

      END; { ELSE - not stuck }
  stucklast := stucknow
UNTIL (near OR stuck OR (cycle >= maxcycles));

WRITELN(F);
WRITELN(F,'DFP METRIC:');
WriteMat(F,H,lastb);

END; {DFP}
```