The Asypow S(plus) Library for Asymptotic Power Calculations

BARRY W. BROWN, JAMES LOVATO, and KATHY RUSSELL

The University of Texas M. D. Anderson Cancer Center, 1515 Holcombe Boulevard, Houston, TX 77030. This work was supported by Cancer Center Core grant CA16672 from the National Cancer Institute, a cooperative study agreement with IBM, and the personal generosity of Pat and Larry McNeil.

How to Obtain the Routines Described

Software written by members of the section is freely available to anyone. Readers with Internet access and a browser might note the following web site addresses:

University of Texas M. D. Anderson Cancer Center Home Page:
http://utmdacc.mdacc.tmc.edu/

Department of Biomathematics Home Page:
http://odin.mdacc.tmc.edu/

Available Software:
http://odin.mdacc.tmc.edu/anonftp/

Our code can also be obtained by anonymous ftp to odin.mdacc.tmc.edu. The index is on file ./pub/index. The index can be viewed by issuing the following Unix command:

finger software@odin.mdacc.tmc.edu

Our code is eventually posted to statlib, which can be accessed accessed at:

http://lib.stat.cmu.edu/

See the S subdirectory.

Legalities

The authors wish to make this code as widely available as possible and hence place no restriction on its copying or use. The authors would appreciate appropriate acknowlegement of the use or incorporation of this code in other packages. This code does incorporate code that is copyrighted to the ACM, for which there are restrictions on commercial distribution

The Fortran routine, **gratio** and those routines called by it are from the following reference:

DiDinato, A. R. and Morris, A. H. "Computation of the incomplete gamma function ratios and their inverse." *ACM Trans. Math. Softw.* 12 (1986), 377-393.

The zero finding routines, **dstinv** and **dstzr** are transliterations from Algol to Fortran of algorithm R of the following reference:

Bus, J. C., and Dekker, T. J. "Two efficient algorithms with guaranteed convergence for finding a zero of a function." *ACM Tran. Math. Softw.* 1(1975), 330-345.

TOMS has the following policy on dissemination of their algorithms:

> Submittal of an algorithm for publication in one of the ACM Transactions implies that unrestricted use of the algorithm within a computer is permissible. General permission to copy and distribute the algorithm without fee is granted provided that the copies are not made or distributed for direct commercial advantage. The ACM copyright notice and the title of the publication and its date appear, and notice is given that copying is by permission of the Association for Computing Machinery. To copy otherwise, or to republish, requires a fee and/or specific permission.

Krogh, F. "Algorithms Policy." *ACM Tran. Math. Softw.* 13(1987), 183-186.

1 Introduction

The asypow library consists of routines written in the S language that calculate power and related quantities utilizing asymptotic methods. A paper describing these methods with examples is in preparation [1]. Two methods are available. The likelihood ratio method (LR) is described in [2]. Another general method appears recently in [3]; and we designate it the SMO method after the initials of the authors.

Here we outline the overall steps in the use of asymptotic sample size and power calculations.

1. The statistician specifies a complete parametric alternative hypothesis model. The specification includes the form of the model model, values of its parameters and the design. If the model involves the comparison of a single parameter over multiple groups, then the design consists of the proportion of subjects in each group. In more complex cases, covariates are involved. If the experimenter controls the covariate values, the design consists of these values and the relative number of observations at each. In an observational study in which the covariate values are not under experimental control, a probability distribution of covariate values must be specified.

2. The designer specifies the constraints on the model parameters that transform the alternative hypothesis into the null hypothesis. Each constraint either sets a parameter value to a constant or it posits the equality of several parameters without specifying their common value. Let the number of constraints be $\mathcal{C}$.

   If there are equality constraints, the model must be reparameterized so that there are only constant constraints. This reparameterization is achieved by replacing each set of parameters, equal according to the null hypothesis, by their differences and by the value of one of them. Fortunately, this process can be automated.

   Both methods described use the model from step 1 and the constraints from step 2 to compute the noncentrality parameter, η, of a noncentral χ^2 distribution that describes the asymptotic distribution of the likelihood ratio under the alternative hypothesis.

3. The critical value, V, is computed from the significance level, α, from $1-\alpha = \chi^2_{\mathcal{C}}(V)$ where $\chi^2_{\mathcal{C}}(.)$ is the central χ^2 cumulative distribution function with $\mathcal{C}$ degrees of freedom. Power is the probability that a random variable distribued as a noncentral χ^2 with $\mathcal{C}$ degrees of freedom and noncentrality parameter, η, exceeds V. If sample size is to be calculated, an iterative calculation finds the requisite noncentrality parameter.

2 Documentation/Help

The asypow routines, unfortunately, are inherently more difficult to use than many others in Splus. This is a result of the need to specify a comparatively large amount of information for each calculation.

To mitigate this problem, a cheatsheet and an online routine, asypow.help, are available as well as the traditional help files. The routine, asypow.help prints most of the text available in the cheatsheet as items chosen from a menu.

3 Methods

3.1 Likelihood Ratio (LR) Method

(This presentation follows Cox and Hinkley [2, Section 9.3].) Let $\mathbf{y} = \{y_1, \ldots, y_n\}$ be a realization of independent, identically distributed random variables,$\{Y_1, \ldots, Y_n\}$, having a density $f(y;\theta)$, where θ is a vector of unknown parameters. The log likelihood of the observations, $\mathbf{y}$, at the parameter value θ is

$$l(\mathbf{y};\theta) = \sum_{s=1}^{n} \log(f(y_s;\theta)). \tag{1}$$

Let $\tilde{\theta}$ denote the alternative hypothesis value of θ. The expected information matrix $\mathbf{i}$ is defined by

$$\mathbf{i}_{j,k} = -\sum_{s=1}^{n} \mathcal{E}_{\tilde{\theta}} \left(\frac{\partial^2 l(Y_s;\theta)}{\partial\theta_j \partial\theta_k} \right). \tag{2}$$

This expectation can be written as

$$\mathcal{E}_{\tilde{\theta}} \left(\frac{\partial^2 l(Y_l;\theta)}{\partial\theta_j \partial\theta_k} \right) = \int \frac{\partial^2 l(\mathbf{y};\theta)}{\partial\theta_j \partial\theta_k} f(y;\tilde{\theta}) dy, \tag{3}$$

where the integration is over the set of possible values of Y. Note that the expectation is calculated at the assumed true value of θ. If the Y_i assume discrete values, $f(y|\theta)$ is the probability that $Y = y$, and the above integral is replaced by a sum over all possible outcomes, y.

The presence of covariates in the model requires another level of computation to obtain the information matrix. When the experimenter chooses covariate values, the information matrix is computed separately for each value, and the results are summed. In observational studies in which covariates are random, the information matrix is integrated, component by component, over the population covariate distribution.

The null hypothesis determines the values of specific components of θ, which is partitioned into (ψ, λ), where the first set of parameter values, ψ, have been

set by the null hypothesis to the value ψ_0. The remaining parameters, λ, are not constrained by the null hypothesis.

The expected information matrix, $\mathbf{i}$, is partitioned in correspondence to θ into

$$\begin{pmatrix} \mathbf{i}_{\psi\psi} & \mathbf{i}_{\psi\lambda} \\ \mathbf{i}_{\lambda\psi} & \mathbf{i}_{\lambda\lambda} \end{pmatrix}.$$

Let $\hat{\theta}$ be the maximum likelihood estimate of θ, $\hat{\theta}_0$ be the value that maximizes the likelihood subject to $\psi = \psi_0$, and $\tilde{\psi}$ be the presumed true value of ψ. The results central to LR power and sample-size calculations are as follows:

The asymptotic distribution of $2[l(\hat{\theta}) - l(\hat{\theta}_0)]$ is noncentral χ^2, $\chi^2_{\mathcal{C}}(x|\eta)$ with noncentrality parameter

$$\eta = \left[\tilde{\psi} - \psi_0\right]^T \left[\mathbf{i}_{\psi\psi} - \mathbf{i}_{\psi\lambda}\mathbf{i}_{\lambda\lambda}^{-1}\mathbf{i}_{\lambda\psi}\right] \left[\tilde{\psi} - \psi_0\right] \tag{4}$$

and degrees of freedom, $\mathcal{C}$ equal to the number of elements in ψ_0, i.e., the number of constraints.

3.2 SMO Method

This method is named with the initials of its authors, [3]. The method is easily motivated from the assumption that asymptotically, $2(l(\mathbf{y};\hat{\theta}) - l(\mathbf{y};\hat{\theta}_0))$ is distributed as $\chi^2_{\mathcal{C}}(x|\eta)$, as per the likelihood ratio calculations but with a different value of η. The model parameters are again partitioned into sets constrained and unconstrained by the null hypothesis. Let the partitioning yield $\tilde{\theta}_0 = (\psi_0, \tilde{\lambda}_0)$; $\tilde{\lambda}_0$ is that value of λ that maximizes $\mathcal{E}_{\tilde{\theta}}(l(Y|(\psi_0, \lambda))$.

For large n, $\hat{\theta} \to \tilde{\theta}$ and $\hat{\theta}_0 \to \tilde{\theta}_0$, so

$$2\mathcal{E}_{\tilde{\theta}}\{l(\mathbf{y};\hat{\theta}) - l(\mathbf{y};\hat{\theta}_0)\} \to 2\mathcal{E}_{\tilde{\theta}}\{l(\mathbf{y};\tilde{\theta}) - l(\mathbf{y};\tilde{\theta}_0)\} \tag{5}$$

The expectation of the non-central χ^2 is $\mathcal{C} + \eta$, so by equating expectations, we have the approximation

$$\eta = 2\mathcal{E}_{\tilde{\theta}}\{l(\mathbf{y};\tilde{\theta}) - l(\mathbf{y};\tilde{\theta}_0)\} - \mathcal{C} \tag{6}$$

SMO used an expansion around the null hypothesis parameter values that produced a term involving first and second derivatives of the expected log likelihood. They found that in practice, this term nearly cancels $\mathcal{C}$ above, and arrive at the above formula without this term.

$$\eta = 2\mathcal{E}_{\tilde{\theta}}\{l(\mathbf{y};\tilde{\theta}) - l(\mathbf{y};\tilde{\theta}_0)\} \tag{7}$$

Equations (6) and (7) above give two means of estimating the noncentrality parameter. The more conservation includes the degrees of freedom term $\mathcal{C}$.

In simple cases, the value of $\tilde{\lambda}_0$ is immediate, but in more complex cases it must be determined through nonlinear maximization. The LR method implicitly uses an asymptotic approximation to $\tilde{\lambda}_0$, a value which usually does not maximize the log likelihood under the null hypothesis. This may be one reason that the SMO method offers superior performance in some cases.

4 Example: Single Sample Poisson

PROBLEM: What sample size is required to reject with power 0.8 the null hypothesis that a Poisson mean is $\lambda = 2.0$ when $\lambda = 3.0$ using a two-sided test with 0.05 significance level?

4.1 Noncentrality Parameter: LR

The only parameter of the model is the mean, λ, so the information matrix is a scalar. The Poisson density is

$$f(y) = \frac{(\lambda^y) * exp(-\lambda)}{y!}.$$

For a single observation, the log likelihood is

$$l(y;\lambda) = y * \log(\lambda) - \lambda - \log(y!).$$

The second derivative of the log likelihood with respect to lambda is

$$\frac{\partial^2 l(y;\lambda)}{\partial \lambda^2} = \frac{-y}{\lambda^2}.$$

Since the number of events, y, is distributed Poisson the expectation of y is λ, the information matrix is

$$i = \frac{1}{\lambda}.$$

The expected information for a single observation is $\frac{1}{3} = 0.33$, so for n observations, the expected information will be $0.33n$. Using (4), the noncentrality parameter for a null hypothesis value of λ_0 is $(\lambda_0 - 3.0)^2 * 0.33n$, which at $\lambda_0 = 2.0$ is $0.33n$.

4.2 Noncentrality Parameter: SMO

The Poisson density is

$$f(y;\lambda) = \frac{(\lambda^y) * exp(-\lambda)}{y!}.$$

For a single observation, the log likelihood is

$$l(y;\lambda) = y * \log(\lambda) - \lambda - \log(y!).$$

The expected value of the log likelihood of λ_a, the alternative hypothesis value of λ over λ_0 is given by the Poisson density at λ_a times the log likelihood at λ_0.

$$\mathcal{E}_{\lambda_a}(l(y;\lambda_0)) = \sum_{y=0}^{\infty} f(y;\lambda_a) l(y;\lambda_0)$$

$$= \sum_{y=0}^{\infty} (y * \log(\lambda_0) - \lambda_0 - \log(y!)] * f(y;\lambda_a)$$

$$= \log(\lambda_0)(\sum_{y=0}^{\infty} y f(y;\lambda_a)) - \lambda_0(\sum_{y=0}^{\infty} f(y;\lambda_a)) - (\sum_{y=0}^{\infty} \log(y!) f(y;\lambda_a))$$

NOTES:

- $\sum_{y=0}^{\infty} y * f(y; \lambda_a)$ is the expected value of y, i.e., λ_a.
- $\sum_{y=0}^{\infty} f(y; \lambda_a)$ is the sum of the density of y, i.e., 1.
- Let $K = -\sum_{y=0}^{\infty} \log(y!) * f(y; \lambda_a)$.

Then,

$$\mathcal{E}_{\lambda_a}(l(y; \lambda_0)) = \log(\lambda_0) * \lambda_a - \lambda_0 + K.$$

The most conservative SMO methods for estimating the noncentrality parameters for n observations shown in (6) is

$$2(\mathcal{E}_{\lambda_a}(l(y; \lambda_a)) - \mathcal{E}_{\lambda_a}(l(y; \lambda_0)))n - \mathcal{C}.$$

which is

$$= 2(\log(\lambda_a)\lambda_a - \lambda_a + K - \log(\lambda_0)\lambda_a + \lambda_0 - K)n - 1$$

$$= 2 * (\lambda_a \log(\frac{\lambda_a}{\lambda_0}) - \lambda_a + \lambda_0)n - 1.$$

Hence, the noncentrality parameter for a null hypothesis value of $\lambda_0 = 2.0$ and alternative hypothesis of $\lambda = 3.0$ is $.43n - 1$.

A less conservative methods given in (7) does not include the degrees of freedom term, $\mathcal{C}$, and the noncentrality parameter is $.43n$.

4.3 Requisite Noncentrality Parameter

We will reject the null hypothesis if twice the LR is greater than the 0.95 quantile of the (central) χ_1^2 distribution, i.e., 3.84. The probability of exceeding 3.84 is

$$1 - \chi_1^2(3.84|\nu)$$

where $\chi_d^2(x|\nu)$ is the cumulative distribution function of the noncentral chi-square distribution with d degrees of freedom and noncentrality parameter ν. The value of ν making the probability of exceeding 3.84 equal to 0.8 is 7.85.

4.4 Sample Size

Equating the noncentrality parameters to 7.85, we obtain $n = 24$ for the LR method and $n = 20$ for the more conservative SMO method or $n = 18$ for the less conservative SMO method.

5 The Routines

The asypow library provides routines for calculating the sample size, power or significance level for hypothesis testing. The available asypow routines can be divided into four categories:

1. **No-Covariate Models:** Single or multiple groups without covariates. The names of these routines end in the characters, **'.kgp'**.

Routine Name	Outcome Type
binomial.kgp	Binomial
poisson.kgp	Poisson
expsurv.kgp	Exponential Survival
ordinal.kgp	Ordinal
multinomial.kgp	Multinomial

2. **Single-Covariate Models:** Single or multiple groups using models involving a single covariate, which may occur either only in a linear term or in a linear and quadratic term. The names of routines considering covariate values end in the characters, **'.design'**.

Routine Name	Outcome Type
binomial.design	Binomial
poisson.design	Poisson
expsurv.design	Exponential Survival
ordinal.design	Ordinal

3. **Multivariate Models:** Single or multiple groups using models with multiple covariates. These routines are primarily used for tabulated data.

Routine Name	Model Type
mvlogistic.design	Logistic
mvloglin.design	Log-Linear

4. **Generic (User Specified) Models:**

Routine Name	Method
generic.model.lr	Likelihood Ratio
generic.model.smo	SMO

6 Use of the Routines

6.1 No-Covariate Models

The following steps explain the use of the asypow routines for the models that do not consider a covariate, (**binomial.kgp**, **poisson.kgp**, **expsurv.kgp**, **multinomial.kgp**, **ordinal.kgp**). A typical calling sequence for a prototype model is shown below where <model> is replaced with the name of the model being used.

```
<model>.kgp(method,
            <theta.ha>,group.size=1,
            const.constraint.group,
            const.constraint.param,
            const.constraint.value,
            equal.constraint.group,
            equal.constraint.param,
            significance,power,sample.size,
            smo.df=T)
```

The argument <theta.ha> is named after the parameters of the distribution, for example, p.ha for the binomial distribution and lambda.ha for the poisson distribution. Some routines have extra options. The exact calling sequence and the model parameters for each routine are available in the individual help files.

Each routine returns a list containing all values used in the construction of the problem. Printing the return value produces a reconstruction of the original problem. The various components of the list can be identified using the **'names()'** function.

6.1.1 Choose a method.

The two asymptotic methods discussed are available for calculations. To choose one, set the argument method to "lr" for the likelihood ratio method or "smo" for the SMO method.

6.1.2 Set the alternative hypothesis.

Define the parameter values of the alternative hypothesis. For example in the **binomial.kgp** routine the parameter is the probability of event, p. If the value of p under the alternative hypothesis is 0.4, then set p.ha to 0.4.

```
> p.ha <- 0.4
```

Each model allows for more than one treatment group. If there is more than one treatment group in the model, the parameter values must be set for each group. This is done by making the argument a matrix with a row for each group. For example, if there were three groups in the binomial models and the value of p was 0.4 for the first group, 0.3 for the second group and 0.9 for the third group p.ha would be set to a column vector.

```
> p.ha <- rbind(0.4,0.3,0.9)
```

6.1.3 Set the proportions of subjects in each group.

The routines allow for unequal sample sizes for models with more than one group. If this is the case then set the argument **group.size** to be a vector of the relative sample sizes. The value of the i^th component is the relative sample size of the i^th group. If this value is specified, it should be a vector whose length is the same as the number of rows in <theta.ha>, the alternative hypothesis parameters.

For example, in the three group binomial model above if 50% of the subjects are assigned to group 3 while 25% are assigned to group 1 and the final 25% to group 2, then **group.size** would be

```
> group.size <- c(.25,.25,.5).
```

If the model has only one group or there are equal sample sizes in each group then the argument **group.size** does not need to be set.

6.1.4 Specify constraints on the parameters that transform the alternative hypothesis into the null hypothesis.

There are two types of constraints. Constant constraints set a parameter to a specific value and equality constraints posit the equality of two or more parameters.

6.1.5 Setting constant constraints.

Constant constraints are defined by three values, a character description of the parameter, the group number of the parameter, and the value the parameter is being set to. These are set in the arguments:

- **const.constraint.group**
- **const.constraint.param**
- **const.constraint.value**

If more than one constant constraint is being defined then the above arguments should be vectors of the same length with the i^{th} component defining the i^{th} constraint.

Each model has different parameters available

Routine	Parameter Names	Meaning
binomial.kgp	p	The probability of an event.
poisson.kgp	lambda	The mean.
expsurv.kgp	rate	The exponential failure rate.
multinomial.kgp	p1,p2,...,p(r-1)	The probabilities for r-1 categories.
ordinal.kgp	p1,p2,...,p(r-1)	The probabilities for r-1 categories.

Note that for the multinomial and ordinal model with r categories, there are only r-1 parameters. This is because the probabilities of the categories must add to one, hence the final probability is redundant. For a more detailed description of the model parameters available see the individual help files.

The arguments **const.constraint.group** and **const.constraint.param** do not need to be specified if there is only one group or one parameter in the model respectively. If the const.constraint.* arguments are not specified then there are no constant constraints.

For example, if the null hypothesis in a one group binomial is to set the parameter 'p' to 0.5 then the argument values would be set in the following fashion

```
> const.constraint.group <- 1
> const.constraint.param <- "p"
> const.constraint.value <- 0.5
```

Since there is only one group and one parameter value in the model the values for **const.constraint.group** and **const.constraint.param** do not need to be set.

For another example, consider a 5 category multinomial model with 2 groups. In this case the available parameter are "p1", "p2", "p3" and "p4". Assume we want to set the values in group 1 to (0.1,0.2,0.3,0.3) and in group 2 to (0.2,0.2,0.3,0.2).

```
> const.constraint.group <- c(1,1,1,1,2,2,2,2)
> const.constraint.param <- c("p1","p2","p3","p4","p1","p2","p3","p4")
> const.constraint.value <- c(0.1,0.2,0.3,0.3,0.2,0.2,0.3,0.2)
```

6.1.6 Setting equality constraints.

Equality constraints are defined by the character description and group number of each parameter being set equal to the others. The group number is set in the vector **equal.constraint.group** and the character description is set in the vector **equal.constraint.param**.

These arguments do not need to be specified if there is only one group or only one parameter in the model respectively. If the **equal.constraint.*** arguments are missing then there are no equality constraints in the model.

For example, to set the probabilities equal in a two group binomial model set

```
> equal.constraint.group <- c(1,2).
```

The argument **equal.constraint.param** is not required because there is only one parameter in the model.

To set all the probabilities equal in a one group, five category multinomial model set (Recall the parameters are "p1",...,"p4").

```
> equal.constraint.param <- c("p1","p2","p3","p4")
```

The argument **equal.constraint.group** is not required because there is only one group in the model.

For more than one equality constraint, the arguments **equal.constraint.param** and **equal.constraint.group** should be a list of vectors where each element in the list defines an equality constraint. For example, in a three category ordinal model the parameters are the category probabilities "p1" and "p2". Suppose the model has two groups and the null hypothesis is that the category probabilities in the 2 groups are the same, then set

```
> equal.constraint.param <- list(c("p1","p1"),c("p2","p2"))
> equal.constraint.group <- list(c(1,2),c(1,2)).
```

6.1.7 Entering constraints at the terminal.

If all five constraint arguments:

const.constraint.param **const.constraint.group**
const.constraint.value **equal.constraint.group**
equal.constraint.value

are missing, then the routines will query the user to enter the constraints interactively. This is often easier for the user.

6.1.8 Set the significance level, power and sample size.

The <model>.kgp routines can calculate the significance level, power or sample size for a hypothesis test given the other two values. The user should enter values for two of the three arguments: **significance, power** or **sample.size**. The third will be calculated by the routine. To make multiple calculations at once the values of **significance, power** or **sample.size** can be vectors. The missing argument can be specified by a -1 or not sending argument to the function.

For example, to calculate the sample size at a power of 0.8 and significance level of 0.01, 0.05 and 0.1 set

```
> significance <- c(0.01,0.05,0.1)
> power <- 0.8
```

6.1.9 Set the degrees of freedom option for the SMO method.

The final argument for the routines is **smo.df**. This is a logical variable which is only relevant if method is "smo". If **smo.df** is TRUE sample size is calculated via the equation $\eta = 2\mathcal{E}_{\tilde{\theta}}\{l(\mathbf{y};\tilde{\theta}) - l(\mathbf{y};\tilde{\theta}_0)\} - \mathcal{C}$ where η is the non-centrality parameter of the Chi-Square model and $\mathcal{C}$ is the degrees of freedom. This is the most conservative calculation. If FALSE, sample size is calculated via the equation $\eta = 2\mathcal{E}_{\tilde{\theta}}\{l(\mathbf{y};\tilde{\theta}) - l(\mathbf{y};\tilde{\theta}_0)\}$. By default, the more conservative calculation is made.

6.2 Univariate Design Models

Popular likelihood based models posit that the parameter of the distribution depends on a covariate, x, through design parameters, a,b,c,.... A linear combination of the covariates is widely used to model this dependence. For example, let

$$u = a + bx + cx^2$$

The asypow routines allow for a linear or quadratic combination. The function which ties u to the value of the parameter is called a link function. Different link functions are appropriate depending on the model.

The following steps explain the use of the asypow routines for design models, (**binomial.design**, **poisson.design**, **expsurv.design**, **ordinal.design**). A typical calling sequence for a designed models is show below where <model> is replaced with the name of the model being used.

```
<model>.design(method,
               theta.ha,design="linear",link,
               xpoints, natx=1, group.size=1,
               const.constraint.group,
               const.constraint.param,
               const.constraint.value,
               equal.constraint.group,
               equal.constraint.param,
               significance,power,sample.size,
               smo.df=T)
```

Some routines have extra options. The exact calling sequence and the model parameters for each routine are available in the individual help files.

6.2.1 Choose a method.

The argument method functions the same in the design routines as the no-covariate routines to select the likelihood ratio or SMO method for the calculations.

6.2.2 Select a design.

The parameter design specifies the function of the covariate x that will be used. Linear indicates, $u = a + bx$, and quadratic indicates, $u = a + bx + cx^2$. Design is "linear" for a linear combination and "quadratic" for a quadratic combination.

Note: For the ordinal model with r categories a linear design specifies $u[i] = a[i]+bx$ and a quadratic design specifies $u[i] = a[i]+bx+cx^2$ for $i = 1, \ldots, r-1$. The model assumes a different intercept term for each category but the other regression terms are the same for all categories.

6.2.3 Select a link function.

The parameter link specifies the function which transforms the linear combination of the covariate into the distribution parameters. The available link functions depend on the model.

Routine	Links Available	Link Formula
binomial.design	logistic	$p = \frac{\exp(u)}{(1+\exp(u))}$
	complementary log	$p = 1 - \exp(-\exp(u))$
poisson.design	exponential	$p = \exp(u)$
expsurv.design	exponential	$rate = \exp(u)$
ordinal.design	logistic	$p[i] = \frac{\exp(u[i])}{(1+\exp(u[i]))}$
	complementary log	$p[i] = 1 - \exp(-\exp(u[i]))$

The link is a parameter only in the routines **binomial.design** and **ordinal.design**, where there is more than one available.

6.2.4 Set the alternative hypothesis.

Define the parameter values at the alternative hypothesis. For the binomial, poisson and clinical trial with exponential survival models the parameters of the model are "a" and "b" for a linear design and "a", "b" and "c" for a quadratic design. For an r category ordinal design the parameters of the model are "a1","a2",... up to the number of categories minus one, plus "b" for a linear design or "b" and "c" for a quadratic design.

The value of these parameters under the alternative hypothesis are defined in the argument **theta.ha**. **theta.ha** is a vector with the values of the parameters under the null hypothesis. For example, in a linear binomial design where a is 1 and b is .3 set

```
> theta.ha <- c(1,.3).
```

For a 3 category quadratic ordinal design where the parameters are a1,a2,b and c giving rise to the following linear combinations

$$u1 = a1 + bx + cx^2$$

$$u2 = a2 + bx + cx^2$$

suppose a1 is .1, a2 is .2, b is .3 and c is .4. Set

```
> theta.ha <- c(.1,.2,.3,.4).
```

Each model allows for more than one treatment group. If there is more than one treatment group in the model, the parameter values must be set for each group. This is done by making the argument a matrix with a row for each group. For example, **theta.ha** in a three group linear binomial model might be

```
> theta.ha <- rbind(c(1,.3),c(2,.4),c(3,.5)).
```

6.2.5 Set the covariate values.

The argument **xpoints** is a vector containing the covariate values in the trial. If there is more than one group in the trial and each group has different covariate values, then **xpoints** is a matrix where each row represents a group. In this case, **xpoints** must have the same number of rows as **theta.ha**.

6.2.6 Set the number of observations at each covariate value.

The routines allow for unequal sample sizes at each covariate value in the design. If this is the case then define the argument **natx** so that natx[i,j] is the relative sample size at covariate xpoints[i,j].

For example, if a trial has 2 covariate values defined in **xpoints** as

```
> xpoints <- c(1,2)
```

and 60% of the subjects have covariate value 1 while 40% of the subjects have covariate value 2 then set **natx**

```
> natx <- c(0.6,0.4)
```

6.2.7 Set the percentage of subjects in each group.

The routines allow for unequal sample sizes for models with more than one group. If this is the case then set the argument **group.size** to be a vector of the relative sample sizes. The argument **group.size** functions the same in the design routines as the no-covariate routines.

6.2.8 Specify constraints on the parameters that transform the alternative hypothesis into the null hypothesis.

The constraints are defined by the arguments

const.constraint.param	**const.constraint.group**
const.constraint.value	**equal.constraint.group**
equal.constraint.value	

These arguments are the same as for the no-covariate routines.

6.2.9 Set the significance level, power and sample size.

The <model>.design routines can calculate the significance level, power or sample size for a hypothesis test given the other two values. The user should enter values for two of the three arguments: **significance, power** or **sample.size**. The third will be calculated by the routine. These arguments function the same in the design routines as the no-covariate routines.

6.2.10 Set the degrees of freedom option for the SMO method.

The final argument for the routines is **smo.df** which determines if the degrees of freedom are used in the SMO calculations. This argument functions the same in the design routines as the no-covariate routines.

6.3 Multivariate Design Models

The asypow library provides two functions which provide a binomial design where the probability of an event is a function of k multiple covariates, x[1],...,x[k] via a combination of coefficient coef[1],...,coef[k].

In the multivariate logistic design (**mvlogistic.design**) the probability of an event is a logistic function of u where $u = x[1]coef[1] + \cdots + x[k]coef[k]$.

In the multivariate log-linear design (**mvloglin.design**) the probability of an event is an exponential function of u where $u = \log(coef[1])x[1] + \cdots + \log(coef[k])x[k]$.

The usual use of these routines is for tabulated data in which case the x's will all be 0 or 1 valued indicator variables.

The following steps explain the use of the asypow routines for multivariate design models, **mvlogistic.design** and **mvloglin.design**. A typical calling sequence for a designed models is show below where <model> is replaced with the name of the model being used, either logistic or loglin for log-linear.

```
mv<model>.design(method,
                 coef.ha, xpoints, rss=1,
                 const.constraint.param,
                 const.constraint.value,
                 equal.constraint.param,
                 significance,power,sample.size,
                 smo.df=T)
```

Use of these routines is similar to using other asypow routines.

6.3.1 Choose a method.

This argument method functions the same in the design routines as the no-covariate routines to select the likelihood ratio or SMO method for the calculations.

6.3.2 Set the alternative hypothesis.

The parameters for multivariate designs are named "coef1", "coef2",... up to the number of covariates in the model. The value of these parameters under the null hypothesis are defined in the argument **coef.ha**.

6.3.3 Set the design points.

The design points are the values of the k covariates included in the trial. These are set in the arguments **xpoints**. **xpoints** is a matrix of dimension $(n \times k)$, each row of which gives values of the k covariates at one of the n design points.

Note: Most models will include a constant term and the column of **xpoints** corresponding to this term will be identically 1.

For example, if there are three covariates x1, x2, and x3 and three design points (x1,x2,x3) = {(1,1,0),(1,0,1) or (1,1,1)} then **xpoints** would be

```
> xpoints <- rbind(c(1,1,0),c(1,0,1),c(1,1,1)).
```

6.3.4 Set the proportion of subjects at each design point.

The routines allow for unequal sample sizes at each design point in **xpoints**. If this is the case then the define the argument **rss** so that rss[i] is the relative sample size at the design point defined in row i of **xpoints**.

For example, if for the three design points defined in **xpoints** above we put 50% of the subject at design point 1 and 25% of the subjects at design points 2 and 3 rss would be

```
> rss <- c(0.5,0.25,0.25).
```

6.3.5 Specify constraints on the parameters that transform the alternative hypothesis into the null hypothesis.

The constraints are defined by the arguments:

- **const.constraint.param**
- **const.constraint.value**
- **equal.constraint.param**.

There are no group indicators since these models are not programmed to consider multiple groups. Setting constraints works the same as it does with other asypow routines.

6.3.6 Set the significance level, power and sample size.

The mv<model>.design routines can calculate the significance level, power or sample size for a hypothesis test given the other two values. The user should enter values for two of the three arguments: **significance, power** or **sample.size**. The third will be calculated by the routine. These arguments function the same in the design routines as the no-covariate routines.

6.3.7 Set the degrees of freedom option for the SMO method.

The final argument for the routines is **smo.df** which determines if the degrees of freedom are used in the SMO calculations. This argument functions the same in the design routines as the no-covariate routines.

6.4 Generic Models using the Likelihood Ratio Method

The function **generic.model.lr** allows the user to calculate the significance level, power or sample size for a hypothesis test using the likelihood ratio method for a model not provided by the asypow library. The calling sequence of this routine is shown below.

```
generic.model.lr(theta.ha,info.mat,
                 const.constraint.param,
                 const.constraint.value,
                 equal.constraint.param,
                 significance,power,sample.size)
```

This routine works similar to other asypow library routines. The parameters are named "theta1", "theta2", ...

The user must provide the information matrix for the model. The information matrix is the second derivate matrix of the expected log likelihood under the alternative hypothesis, the negative of the hessian matrix. The information matrix should be scaled to be the information for one observation.

6.5 Generic Models using the SMO Method

The function **generic.model.smo** allows the user to calculate the significance level, power or sample size for a hypothesis test using the SMO method for a model not provided by the asypow library. The calling sequence of this routine is shown below.

```
generic.model.smo(theta.ha,theta.lo,theta.hih,
                  expect.loglike,
                  const.constraint.param,
                  const.constraint.value,
                  equal.constraint.param,
                  significance,power,sample.size,
                  smo.df=T)
```

This routine works similar to other asypow library routines. The parameters are named "theta1", "theta2", ...

The user must provide the arguments **theta.lo** and **theta.hi**: **theta.lo** is an array of lower limits on the parameters, and **theta.hi** is an array of upper limits on the parameters.

The user must also provide an S function that calculates the expected log-likelihood of the model assuming that parameter values, **theta.ha**,are correct at a parameter point, **theta.ho**. This function is passed in the argument **expect.loglike**. **expect.loglike** is a function of the two vectors: **theta.ha** and **theta.ho**.

If density(theta,...) is a function which calculates the density of the model at the parameter values theta, and loglike(theta,...) is a function which calculates the log-likelihood of the model at the parameter values theta, then the expected log-likelihood at **theta.ho** over **theta.ha** is density(theta.ha)*loglike(theta.ho) integrated over the appropriate space.

7 References

1. Brown, Barry W., Lovato, James and Russell, Kathy (1996) "Asymptotic Power Calculations and the Art of the Merely Reasonable" *In Preparation.*
2. Cox, D.R. and Hinkley, D.V. (1974), *Theoretical Statistics.* London: Chapman and Hall.
3. Self, S.G., Mauritsen, R.H. and Ohara, J. (1992) "Power Calculations for Likelihood Ratio Tests in Generalized Linear Models." *Biometrics,* 48, 31-39.