

**FITTING THE
SCHWARTZ-BARRETT-MARSHALL MODEL**

Germán Rodríguez
James Trussell

Office of Population Research
Working Paper No. 97-3

April 1997

Acknowledgement:

This work was supported by NICHD grants R01 HD34016 and P30 HD-AG32030

Fitting the Schwartz-Barrett-Marshall Model*

Germán Rodríguez
James Trussell

September 1996
revised April 1997

These notes summarize work done in fitting the Schwartz-Barrett-Marshall model of conception probabilities. Section 1 derives the results needed for calculating the estimates and their standard errors. Section 2 documents the S functions that actually do the work. Sections 3 to 5 report selected results for the Barrett-Marshall and the Wilcox-Weinberg datasets.

1 The Model and the Estimation Procedure

The data consist of conception and coitus indicators for n cycles. Let $y_i = 1$ if the i -th cycle resulted in conception and let $x_{ij} = 1$ if there was intercourse in day j of cycle i . Conventionally we number the day of ovulation as day 0.

1.1 The Model

The overall probability of conception p_i in cycle i is assumed to be

$$p_i = k(1 - \prod_j (1 - \lambda_j)^{x_{ij}}) \quad (1)$$

where $0 < k < 1$ and $0 < \lambda_j < 1 \forall j$.

With *individual* data we assume that y_i is Bernoulli with parameter p_i . This set-up can be extended to *grouped* data by letting y_i denote the proportion of cycles resulting in conception out of n_i cycles with coital pattern described by the vector x_i .

*This work was supported by NIH Grant R01HD34016.

In both cases the kernel of the log-likelihood can be written as

$$\log L = \sum_i n_i \{y_i \log(p_i) + (1 - y_i) \log(1 - p_i)\} \quad (2)$$

with $n_i = 1 \forall i$ for individual data.

1.2 Maximum Likelihood Estimation

For estimation purposes it will be convenient to work with the *logits* of k and λ_j , so that the parameters are automatically constrained to $(0, 1)$. Let

$$\alpha = \text{logit}(k) \quad \text{and} \quad \beta_j = \text{logit}(\lambda_j) \quad (3)$$

We now consider the derivatives of the log-likelihood function. Using the chain rule:

$$\frac{\partial \log L_i}{\partial \alpha} = \sum_i \frac{\partial \log L_i}{\partial p_i} \frac{\partial p_i}{\partial k} \frac{\partial k}{\partial \alpha} \quad (4)$$

and

$$\frac{\partial \log L_i}{\partial \beta_j} = \sum_i \frac{\partial \log L_i}{\partial p_i} \frac{\partial p_i}{\partial \lambda_j} \frac{\partial \lambda_j}{\partial \beta_j} \quad (5)$$

The derivative of the log-likelihood wrt p_i is

$$\frac{\partial \log L_i}{\partial p_i} = n_i \frac{y_i - p_i}{p_i(1 - p_i)} \quad (6)$$

The derivatives of the p_i wrt k and λ_j are

$$\frac{\partial p_i}{\partial k} = \frac{p_i}{k} \quad (7)$$

and

$$\frac{\partial p_i}{\partial \lambda_j} = \frac{k - p_i}{1 - \lambda_j} x_{ij} \quad (8)$$

The derivatives of k and λ_j wrt α and β_j are, from well-known results for logits,

$$\frac{\partial k}{\partial \alpha} = k(1 - k) \quad (9)$$

and

$$\frac{\partial \lambda_k}{\partial \beta_j} = \lambda_j(1 - \lambda_j) \quad (10)$$

Using these results, the derivatives we need are

$$\frac{\partial \log L_i}{\partial \alpha} = \frac{\partial \log L_i}{\partial p_i} p_i (1 - k) \quad (11)$$

and

$$\frac{\partial \log L_i}{\partial \beta_j} = \frac{\partial \log L_i}{\partial p_i} \lambda_j (k - p_i) x_{ij} \quad (12)$$

These are the values used in the maximization procedure.

1.3 Variance-Covariance Matrix

We are interested in the standard errors of \hat{k} , $\hat{\lambda}_j$ and the products $\hat{k}\hat{\lambda}_j$. In fact, we need the entire variance-covariance matrix of the products $\hat{k}\hat{\lambda}_j$. We propose to estimate these quantities using the *expected* information matrix and the delta method.

Let $\theta = (k, \lambda_1, \dots, \lambda_m)'$ collect the parameters. Then the contribution of the i -th observation to the information wrt two arbitrary parameters θ_u and θ_v is

$$I_i(\theta_u, \theta_v) = E\left(-\frac{\partial^2 \log L_i}{\partial \theta_u \partial \theta_v}\right) = E\left(\frac{\partial \log L_i}{\partial \theta_u} \quad \frac{\partial \log L_i}{\partial \theta_v}\right) \quad (13)$$

where we have used the well-known Bartlett identity.

In our case the expected squared first derivative wrt p_i is

$$E\left[\left(\frac{\partial \log L_i}{\partial p_i}\right)^2\right] = E\left[\frac{n_i^2 (y_i - p_i)^2}{p_i^2 (1 - p_i)^2}\right] = \frac{n_i}{p_i (1 - p_i)} \quad (14)$$

The elements of the expected information matrix are obtained multiplying the expected squared first derivative wrt p_i by the derivatives of p_i with respect to k and λ_j as given in Equations 7 and 8.

For example,

$$I(k, \lambda_j) = \sum_i \frac{n_i}{p_i (1 - p_i)} \frac{p_i}{k} \frac{k - p_i}{1 - \lambda_j} x_{ij} \quad (15)$$

and similarly for $I(k, k)$ and $I(\lambda_u, \lambda_v)$. All these quantities are evaluated at the m.l.e.'s.

The variance-covariance matrix of \hat{k} and $\hat{\lambda}$ is obtained by inverting the information matrix.

To obtain the variance-covariance matrix of the products $\hat{k}\hat{\lambda}_j$ we use the delta method. If $f(\theta)$ is a vector of functions of the parameters θ with derivative matrix D then $\text{var}(f(\hat{\theta})) \approx D\text{var}(\hat{\theta})D'$. In our case the matrix of first derivatives has a simple structure: the derivative of $k\lambda_j$ wrt k is λ_j and the derivative wrt λ_j is k , so

$$\text{var}(\hat{k}\hat{\lambda}) \approx [\lambda, kI] \text{var}(\hat{\theta}) [\lambda, kI]' \quad (16)$$

where λ is the m -vector $(\lambda_1, \dots, \lambda_m)'$, I is the identity matrix of order m , and the right-hand side is evaluated at the m.l.e.'s. It may be useful to note that the variances of the products $\hat{k}\hat{\lambda}_j$, given by the diagonal elements of the matrix in Equation 16, turn out to be

$$\text{var}(\hat{k}\hat{\lambda}_j) \approx \lambda_j^2 \text{var}(k) + k^2 \text{var}(\lambda_j) + 2k\lambda_j \text{cov}(k, \lambda_j) \quad (17)$$

where the right-hand side is evaluated at the m.l.e.'s.

1.4 Analytic Results

The special case where we look at two days in the cycle is of interest because it leads to an explicit solution for the estimates and the standard errors. In this case there are only three possible coital patterns, and the corresponding probabilities of conception are

x_1	x_2	p
1	0	$k\lambda_1$
0	1	$k\lambda_2$
1	1	$k(1 - (1 - \lambda_1)(1 - \lambda_2))$

Since there are three parameters to model three observed proportions, we have an explicit solution. Simple algebra shows that

$$\begin{aligned} \hat{k} &= \frac{p_1 p_2}{p_1 + p_2 - p_3} \\ \hat{\lambda}_j &= p_j / \hat{k} \end{aligned}$$

where $p_j, j = 1, 2, 3$ are the observed proportions.

For days -2,-1 the Barrett-Marshall data (see Section 3) are not consistent with the model and the constrained estimates have $\hat{\lambda}_2 = 1$. For days -1,0, however, the data are consistent with the model and we get

$$\hat{k} = 0.45778, \quad \hat{\lambda}_1 = 0.51737 \quad \text{and} \quad \hat{\lambda}_2 = 0.93620$$

More to the point, $\hat{k}\hat{\lambda}_j = p_j$ for $j = 1, 2$, so the variance of the product can be estimated from the usual binomial formula $p_j(1 - p_j)/n_j$. This gives

	p	se(p)
[1]	0.3972603	0.05727183
[2]	0.2368421	0.04876743

This result will be useful to check the numerical procedures below.

2 The S Functions

2.1 The Data Interface

To simplify fitting models to various datasets and subsets of datasets, we decided to use an *interface* or extractor function of the form `extractor(days)`. This function should return a list with the vector of (proportions of) conceptions y , the matrix of coitus indicators X and a vector of weights w corresponding to the binomial denominators (usually ones).

Here is the interface for the benchmark data:

```
> bm.benchmark
function(days)
{
# Benchmark: returns grouped Barrett-Marshall data for days -1 and 0
# GR 17-June-96
#
  if(length(days) != 2 || any(days != c(-1, 0)))
    stop("Benchmark should be days -1 and 0")
  n <- c(73, 76, 21)
  y <- c(29, 18, 9)/n
  X <- matrix(c(1, 0, 1, 0, 1, 1), 3, 2)
  dimnames(X) <- list(c("1", "2", "3"), c("d1b", "d0"))
  list(y = y, X = X, w = n)
}
```

The function *must* be called with the range -1:0. It returns the proportions of conceptions, coital patterns and number of cycles for the three possible patterns of days -1 and 0.

Interfaces for the other datasets are introduced in Sections 3 and 4.

2.2 Fitting Models

To fit a model we call the function `sbm.fit` defined as follows:

```
> sbm.fit
function(days, extract = bm.extract,
         alpha = logit(0.7), beta = NULL, trace = T)
# Fit the Schwartz-Barrett-Marshall model to a dataset
# using an interface function to retrieve the data.
# GR 11-June-96 rev 17-June-96
#
# extract y vector and X matrix
  days.label <- deparse(substitute(days))
  extract.label <- deparse(substitute(extract))
  data <- extract(days) #
# starting values (based on Royston's estimates)
  if(is.null(beta)) {
    lambda <- c(rep(0.01, 18),
               0.02, 0.03, 0.04, 0.1, 0.2, 0.2,
               0.4, 0.2, rep(0.01, 18))/0.5
    beta <- logit(lambda[sort(days) + 26])
  }
  fit <- ms( ~ sbm.loglik(alpha, beta, data),
            start = list(alpha = alpha,
                        beta = beta), trace = trace) #
# return days, k and daily probs
  fit$days <- days.label
  fit$extract <- extract.label
  fit$k <- antilogit(fit$param[1])
  names(fit$k) <- "k"
  fit$k1 <- antilogit(fit$param[-1]) * fit$k
  names(fit$k1) <- dimnames(data$X)[[2]]
  class(fit) <- c("sbm", class(fit))
  fit
}
```

The function has only one required argument, a range of days, and optional arguments to specify a data interface, starting values for the parameter estimates, and a flag to trace the iterations (true by default).

The first thing the function does is save character versions of the range of days and the name of the data extract function. (These are later stored as attributes of the fit.) The function then calls the user-supplied data extractor.

The next step is to produce starting values. The parameter α is set to the logit of 0.7 unless the user specifies something else. The parameter vector β is set to rounded versions of the estimates in Royston (1982).

The function then calls S's function minimizer `ms`, passing as criterion the function `sbm.loglik`, which is defined below. The `start` parameter tells S that the parameters are α and β and provides starting values.

The call returns an object of class `ms`. We add four new attributes. The first two are the specification of the day range, and the name of the data extraction function. The next two, k and $k\lambda$, convert the parameters from logits to the probability scale. Note that we use the names of the columns of the data matrix X to label the parameter estimates. It is therefore important to make sure that the extraction function assigns dimension names to X .

Finally, we change the class of the object to `sbm` (for Schwartz-Barrett-Marshall) and return. This allows us to define a print method for `sbm` fits:

```
> print.sbm
function(object)
{
# Print method for Schwartz-Barrett-Marshall fits
# GR 17-June-96
#
  cat("log-likelihood:", format( - object$value), "\n")
  cat("estimates (k and k*lambda):\n")
  print(c(object$k, object$kl))
  invisible(object)
}
```

We report the log-likelihood (which is minus the criterion function) and the estimates of k and $k\lambda$.

The criterion function is next:

```
> sbm.loglik
function(alpha, beta, data)
{
# The log-likelihood function for the Schwartz-Barrett-Marshall model
# In this version data is a list with y, X and w (y is proportions)
```

```

# GR 14-June-96 rev 17-June-96
#
  theta <- - log(1 + care.exp(beta))      # log(1-lambda)
  k <- care.exp(alpha)/(1 + care.exp(alpha))
  p <- k * (1 - care.exp(data$X %*% theta)) # prob by cycle
  lambda <- care.exp(beta)/(1 + care.exp(beta)) # prob by day
# log-likelihood
  logLik <- sum(data$w *
    (data$y * log(p) + (1 - data$y) * log(1 - p))) #
# derivatives with respect to p
  den <- p * (1 - p)
  if(any(tiny <- (den < .Machine$double.eps))) {
    warning("Model unstable; fitted probabilities of 0 or 1")
    den[tiny] <- .Machine$double.eps
  }
  der <- (data$w * (data$y - p))/den      #
  ndays <- length(beta)
  gradient <- rep(0, ndays + 1)
  gradient[1] <- sum(der * p * (1 - k))
  for(j in 1:length(beta)) {
    gradient[j + 1] <-
      sum(der * lambda[j] * (k - p) * data$X[, j])
  }
# change signs of logLik and gradient for S minimizer
  logLik <- - logLik
  attr(logLik, "gradient") <- - gradient
  logLik
}

```

This function receives the current parameter estimates and the data. It calculates the fitted values $p = k(1 - \exp\{\eta\})$ where $\eta = X \log(1 - \lambda)$ is a linear predictor. We use `care.exp` rather than `exp` to avoid overflow.

The log-likelihood is then calculated using a direct translation of Equation 2 into S.

Calculation of the derivatives proceeds in two steps. First we calculate the derivative with respect to p using Equation 6. Note that we calculate the denominator first, and hold at the limit very small elements before dividing n by $p(1 - p)$.

The second step involves looping by parameter, multiplying the derivative wrt p by the derivative of p wrt α or β , following Equations 7 and 8.

Finally, we change signs on the log-likelihood and its derivatives, to accommodate the fact that `ms` is a function *minimizer*. The gradient is returned as an attribute of the function, as required by `ms`.

Here's is a run on the benchmark:

```
> bfit <- sbm.fit(-1:0,bm.benchmark)
Iteration: 0 , 1 function calls, F= 110.7504
Parameters:
[1] 0.8472979 1.3862944 -0.4054651
... iterations suppressed ...
Iteration: 15 , 18 function calls, F= 104.992
Parameters:
[1] -0.16918413 1.88113119 0.06942207
> bfit
log-likelihood: -104.992
estimates (k and p=k*lambda):
      k      d1b      d0
0.4577813 0.3972594 0.2368435
```

The procedure converges in 15 iterations. The output agrees with the exact results on page 4.

2.3 Variance-Covariance Matrices

Calculation of standard errors (and more generally variance-covariance matrices) is done in two steps. The function `sbm.covar` calculates the estimated k and λ and their variance-covariance matrix using the results in Section 1.3.

```
> sbm.covar
function(fit)
{
# Calculates variance-covariance matrix of k and lambda from
# a Schwartz-Barrett-Marshall fit
# GR - 16-June-96 rev 17-June-96
#
  if(!inherits(fit, "sbm")) stop(
    "argument must be a Schwartz-Barrett-Marshall fit") #
```

```

# extract parameter estimates
  k <- antilogit(fit$param[1])
  lambda <- antilogit(fit$param[-1])      #
# need X to calculate fitted values!
  days <- eval(parse(, , fit$days))
  extract <- eval(parse(, , fit$extract))
  data <- extract(days)
  theta <- - log(1 + care.exp(fit$param[-1]))
  p <- k * (1 - care.exp(data$X %*% theta))  #
# calculate minus expected second derivative wrt p
  d <- p * (1 - p)
  if(any(tiny <- (d < .Machine$double.eps)))
    d[tiny] <- .Machine$double.eps
  der2 <- data$w/d      # E(dlogLi/dp_i)^2
# calculate derivatives of p wrt k and lambda's
  npar <- ncol(data$X) + 1
  dpdt <- matrix(0, nrow(data$X), npar)
  dpdt[, 1] <- p/k      # der of p wrt k
  for(j in 1:length(lambda))
    dpdt[, j + 1] <- ifelse(data$X[, j],
      (k - p)/(1 - lambda[j]), 0) #
# now build the information matrix for k and lambda
  info <- matrix(0, npar, npar)
  for(i in 1:npar) {
    for(j in i:npar) {
      info[i, j] <-
        sum(der2 * dpdt[, i] * dpdt[, j])
      if(i != j)
        info[j, i] <- info[i, j]
    }
  }
# invert information to get var-cov matrix
  V <- solve(info)
  param <- c(k, lambda)
  names(param) <- c("k", dimnames(data$X)[[2]])
  dimnames(V) <- list(names(param), names(param))
  list(par = param, var = V)
}

```

The argument to the function must be a Schwartz-Barrett-Marshall fit. The function extracts the parameter estimates and converts to the probability scale. To calculate fitted values, however, we need the X matrix. To obtain X we parse the specification of the range of days and the name of the extraction function, and then call it to extract the data again.

The calculation of fitted values and derivatives follows steps very similar to the log-likelihood function, except that instead of first derivatives we use the expected squared first derivatives wrt p . These values are used to build the upper triangle of the information matrix, which is copied to the lower triangle as we proceed.

The variance-covariance matrix is obtained by inverting the expected information. The rest of the code is housekeeping: we collect the estimates of k and λ in a vector and borrow the names of the latter from the columns of the data matrix.

Of course what we really want is the variance-covariance matrix of the probabilities of conception given intercourse on a single day, the $k\lambda_j$. These are calculated using Equation 16, as implemented in the following function

```
> sbm.covkl
function(object)
{
# Calculates variance-covariance matrix of k*lambda
# from a Schwartz-Barrett-Marshall fit
# GR 21-June-96
#
      if(!inherits(object, "sbm"))
        stop("argument must be a Schwartz-Barrett-Marshall fit")
      w <- sbm.covar(object)
      k <- w$par[1]
      lambda <- w$par[-1]      #
# matrix of derivatives of k*lambda wrt c(k,lambda)
      D <- cbind(lambda, k * diag(length(lambda)))
      V <- D %**% w$var %**% t(D)
      kl <- list(par = k * lambda, var = V)
      class(kl) <- "sbmckl"
      kl
}

```

The function checks that the argument is a Schwartz-Barrett-Marshall fit, calls the previous function `sbm.covar` to get the var-cov matrix of k and λ ,

and then uses Equation 16 to calculate the var-cov matrix of $k\lambda$.

The return value is a list with the probabilities (`par`) and their variance-covariance matrix (`var`). The list gets assigned class `sbmckl`. This allows us to define a customized print function:

```
> print.sbmckl
function(object)
{
# Print method for Schwartz-Barrett-Marshall covariance matrix
# GR 21-June-96
#
      se <- sqrt(diag(object$var))
      df <- cbind(object$par, se, object$par/se)
      dimnames(df)[[2]] <- c("kl", "se", "z")
      print(df)
      invisible(NULL)
}
```

This function prints the estimates, standard errors and z-tests side by side. (To print the entire var-cov matrix use `print.default`.) Here are the results on the benchmark:

```
> sbm.covkl(bfit)
           kl           se           z
d1b 0.3972594 0.05727181 6.936386
d0  0.2368435 0.04876753 4.856582
```

The estimates agree exactly with the closed-form results in Section 1.4.

(Note: the functions `sbm.covkl` and `print.sbmckl` supersede the earlier functions `sbm.varkl` and `print.sbmkl` that calculated the variances only.)

3 The Barrett-Marshall Data

The Barrett-Marshall dataset was read from an ascii file,

```
barmarsh.corrected.second.dat
```

in B.Vaughan's `barrett` directory, and saved as an S object `barmarsh`. We named the coitus indicators `d25b` to `d18a` (day 25 before to day 18 after), where day 0 is the last day of hypothermia (day 1 in B-M). There are 103 pregnancies in 2192 cycles.

3.1 The Interface Function

Here is the interface function for the Barrett-Marshall data:

```
>bm.extract
function(days)
{
# Interface function for Barrett-Marshall data frame. Extracts cases
# with intercourse in a given range of days (from -25 to 18) and
# returns a list with conceptions, coitus indicators and weights.
# GR 14-June-96 rev 17-June-96
#
# check days argument
  if(any(days < -25) | any(days > 18))
    stop("Days must be between -25 and 18")
  cols <- sort(days + 27) # day -25 is in column 2
  if(length(cols) > 1 & any(diff(cols) == 0))
    stop("Days must be distinct")      #
# extract y vector and X matrix
  X <- as.matrix(barmarsh[, cols])
  count <- X %*% rep(1, length(cols))
  use <- (count > 0)
  X <- X[use, ]
  y <- barmarsh$conception[use]
  list(X = X, y = y, w = rep(1, length(y)))
}
```

The function checks that the argument evaluates to a set of distinct days in the range (-25,18), picks the coitus indicators from the appropriate columns (day -25 is in column 2), sums by rows to obtain an intercourse count, and uses this count to select cases with coitus in at least one of the days in the range of interest. The conception indicators are picked from a column called conception. The weights are all one, since we have individual data.

Note that the length of `y` is the number of cycles and its sum is the number of conceptions. Here's a simple function to report cycles and conceptions:

```
> sbm.count
function(days, extract)
{
```

```

# Count cycles and conceptions given an interface to a dataset
# GR 20-June-96
#
      data <- extract(days)
      list(cycles = length(data$y), pregnancies = sum(data$y))
}

```

3.2 Selected Fits to Barrett-Marshall Data: Days -9 to 4

Consider first working with days -9 to 4, as used in the paper by Royston (1982). The count of cycles and pregnancies for this period is

```

> sbm.count(-9:4,bm.extract)
$cycles:
[1] 1543

```

```

$pregnancies:
[1] 101

```

To fit the Barrett-Marshall model to this period we used our `sbm.fit` function:

```

> bm94 <- sbm.fit(-9:4,bm.extract)
> bm94 <- sbm.fit(-9:4,bm.extract,alpha=bm96$par[1],beta=bm94$par[-1])

```

The procedure did not converge in 50 iterations. Restarting it from the last set of estimates it converged in 14 iterations. The estimates are:

```

> bm94
log-likelihood: -253.2592
estimates (k and k*lambda):
      k          d9b          d8b          d7b          d6b          d5b
0.5242085 5.429649e-09 0.0001275696 0.0189029 0.04023597 3.431994e-07
      d4b          d3b          d2b          d1b          d0          d1a          d2a
0.1272859 0.1972295 0.1927557 0.3434663 0.1436935 0.06608247 1.233012e-08
      d3a          d4a
0.00508299 0.005683516

```

Note that Royston gets $\hat{k} = 0.38$. We have been unable to reproduce his results. The sum of the estimated daily probabilities $k\lambda_j$ is

```

> sum(bm94$k1)
[1] 1.140547

```

The standard errors are as follows

```
> sbm.covkl(bm94)
      kl          se          z
d9b 5.429649e-09 1.032713e-05 0.0005257658
d8b 1.275696e-04 1.428059e-03 0.0893307729
d7b 1.890290e-02 1.223936e-02 1.5444344049
d6b 4.023597e-02 1.918146e-02 2.0976486860
d5b 3.431994e-07 1.313602e-04 0.0026126594
d4b 1.272859e-01 5.147642e-02 2.4727024953
d3b 1.972295e-01 6.288612e-02 3.1362962361
d2b 1.927557e-01 6.699340e-02 2.8772346560
d1b 3.434663e-01 7.778024e-02 4.4158553326
d0 1.436935e-01 5.752419e-02 2.4979671056
d1a 6.608247e-02 3.521969e-02 1.8762931448
d2a 1.233012e-08 1.672007e-05 0.0007374446
d3a 5.082990e-03 3.931980e-03 1.2927305392
d4a 5.683516e-03 3.641838e-03 1.5606173812
```

Note that we have “holes” in day 5 before ovulation and day 2 after ovulation.

3.3 Selected Fits to Barrett-Marshall Data: Days -5 to 1

Restricting the range to days -5 to 1 with 90 pregnancies in 427 cycles, as done in Schwartz et al. (1980), gives:

```
> bm51<-sbm.fit(-5:1,bm.extract)
> bm51
log-likelihood: -196.248
estimates (k and k*lambda):
      k          d5b          d4b          d3b          d2b          d1b          d0
0.5181773 0.03841258 0.1379178 0.2015542 0.1973683 0.3445385 0.1396421
      d1a
0.06837946
```

with convergence in 18 iterations. The sum of the estimated daily probabilities is

```
> sum(bm51$k1)
[1] 1.127813
```

Finally, here are the standard errors:

```
> bmc51 <- sbm.covkl(bm51)
> bmc51
           kl           se           z
d5b 0.03841258 0.02214568 1.734541
d4b 0.13791776 0.05161766 2.671911
d3b 0.20155420 0.06170863 3.266224
d2b 0.19736829 0.06696376 2.947389
d1b 0.34453853 0.07803795 4.415013
d0  0.13964213 0.05561261 2.510979
d1a 0.06837946 0.03498822 1.954356
```

This time we get no “holes”, and the estimates are identical to those obtained by Schwartz et al (1980). (The complete variance-covariance matrix is listed in an Appendix that summarizes the final five fits of interest.)

4 The Wilcox-Weinberg Data

The second dataset was received by e-mail from Clare Weinberg.

4.1 Documentation

Here is the relevant documentation:

Early Pregnancy Study Data, source NIEHS, 5/96. 740 lines, one per menstrual cycle. Note: Several women were contracepting in their first cycle.

Line number is first, then

W is a weight variable that indicates which cycles have analyzable data. Exclusions were based on unknown conception status, missing intercourse data in the 6-day window ending on the estimated day of ovulation, or no intercourse in the interval I3-I15, where "12" indexes the DoLT, or no estimable day of ovulation. There are 620 with W=1.

PREG is an indicator variable for detected conception There are 199 of these,

EL is an indicator variable for early loss, i.e. a detected conception that ended in loss, where the bleeding began within 6 weeks LMP. There are 48 of these.

I6 - I13 are indicator variables for whether unprotected intercourse was recorded on that day, where I12 corresponds to the estimated day of ovulation. Missing days are indicated by negative numbers.

Note that the ELs on lines 67, 522, and 694 of data were recognized as pregnancies by the women.

Refer questions regarding coding to Clare Weinberg, NIEHS, 919-541-4927, Email WEINBERG@NIEHS.NIH.GOV

In answer to a query about contraceptive use in the first cycle Clare Weinberg stated:

The cycles with contraception will show up as cycles with no intercourse, because what women were asked to record was their days with unprotected intercourse.

We changed the variable names from I6-I13 to d6b-d1a (day 6 before to day 1 after) for consistency, and read the data into a data frame called `weinberg`.

4.2 The Interface Functions

Note that the data have missing values on the coitus days, but selecting on the weight variable `W==1` leads to a set of 620 cycles with complete data. The interface function listed below selects these cases even if the range of interest is less than -6 to 1:

```
> w.extract
function(days)
{
# Interface function for Weinberg data frame. Extracts cases
# with intercourse in a given range of days (from -6 to 1) and
# returns a list with conceptions, coitus indicators and weights.
# GR 20-June-96
#
# check days argument
  if(any(days < -6) | any(days > 1))
    stop("Days must be between -6 and 1")
  cols <- sort(days + 10) # day -6 is in column 4
  if(length(cols) > 1 & any(diff(cols) == 0))
    stop("Days must be distinct")      #
# extract y vector and X matrix
  data <- weinberg[weinberg$W > 0, ]
  X <- as.matrix(data[, cols])
  count <- X %*% rep(1, length(cols))
}
```

```

        use <- (count > 0)
        X <- X[use, ]
        y <- data$PREG[use]
        list(X = X, y = y, w = rep(1, length(y)))
    }

```

Otherwise, this function is very similar to the Barrett-Marshall extractor.

We also needed to fit models using recognized pregnancies only. To accomodate this need, we define a second interface function

```

> wr.extract
function(days)
{
# Interface for Weinberg data, excluding non-recognized pregnancies
# GR 21-June-96
#
# check days argument
  if(any(days < -6) | any(days > 1))
    stop("Days must be between -6 and 1")
  cols <- sort(days + 10) # day -6 is in column 4
  if(length(cols) > 1 & any(diff(cols) == 0))
    stop("Days must be distinct")      #
# use only recognized pregnancies (see documentation)
  data <- weinberg
  data$PREG <- data$PREG - data$EL
  data$PREG[c(67, 522, 694)] <- 1 #
# extract y vector and X matrix
  data <- data[data$W > 0, ]
  X <- as.matrix(data[, cols])
  count <- X %*% rep(1, length(cols))
  use <- (count > 0)
  X <- X[use, ] #
  y <- data$PREG[use]
  list(X = X, y = y, w = rep(1, length(y)))
}

```

The “restricted” interface has three new lines that are used to recode unrecognized pregnancies (EL==1, except for cycles 67, 522 and 694) as 0.

4.3 Selected Fits to Wilcox-Weinberg Data: Days -6 to 1

Here's a fit using all the available (complete) data, with 192 pregnancies in 606 cycles:

```
> w61<-sbm.fit(-6:1,w.extract)
> w61
log-likelihood: -367.1878
estimates (k and k*lambda):
      k          d6b          d5b          d4b          d3b          d2b          d1b
0.3733229 2.064237e-06 0.09935901 0.1552987 0.1368307 0.2747466 0.313319
      d0          d1a
0.3316882 1.907955e-06
```

The procedure converged in 25 iterations. The sum of probabilities is

```
> sum(w61$k1)
[1] 1.311246
```

And here are the standard errors:

```
> wc61 <- sbm.covk1(w61)
> wc61
      k1          se          z
d6b 2.064237e-06 0.0005646581 0.003655729
d5b 9.935901e-02 0.0777015263 1.278726558
d4b 1.552987e-01 0.0652570470 2.379800343
d3b 1.368307e-01 0.0832462980 1.643684922
d2b 2.747466e-01 0.0667782112 4.114315378
d1b 3.133190e-01 0.0570435761 5.492625756
d0 3.316882e-01 0.0858593043 3.863159361
d1a 1.907955e-06 0.0005921054 0.003222323
```

Note that the probabilities of conception are essentially zero for days 6 before and 1 after, and there are no anomalies.

4.4 Selected Fits to Wilcox-Weinberg Data: Days -5 to 0

In view of the previous result, we repeat the fit using days -5 to 0 with 192 pregnancies in 594 cycles:

```

> w50 <- sbm.fit(-5:0, w.extract)
> w50
log-likelihood: -367.1877
estimates (k and k*lambda):
      k      d5b      d4b      d3b      d2b      d1b      d0
0.3733333 0.09935487 0.155297 0.1368277 0.2747404 0.3133121 0.3316633

```

The sum of the daily probabilities $k\lambda_j$ is

```

> sum(w50$k1)
[1] 1.311195

```

Here are the standard errors:

```

> wc50 <- sbm.covk1(w50)
> wc50
      k1      se      z
d5b 0.09935487 0.07769842 1.278724
d4b 0.15529697 0.06525532 2.379836
d3b 0.13682768 0.08324336 1.643707
d2b 0.27474041 0.06677679 4.114310
d1b 0.31331212 0.05704250 5.492608
d0 0.33166334 0.08585581 3.863027

```

Our estimates of the conception probabilities differ from those in Wilcox et al. (1995) by at most 0.002.

Next we repeat the analysis for this range of days using only recognized pregnancies. The ‘restricted’ dataset has 147 pregnancies in 594 cycles.

```

> wr50 <- sbm.fit(-5:0, wr.extract)
> wr50
log-likelihood: -323.0228
estimates (k and k*lambda):
      k      d5b      d4b      d3b      d2b      d1b      d0
0.2878758 1.463137e-16 0.1390712 0.08908828 0.2878758 0.2716303 0.09417483

```

The sum of the daily probabilities is

```

> sum(wr50$k1)
[1] 0.8818403

```

The corresponding standard errors are

```

> wrc50 <- sbm.covkl(wr50)
> wrc50

```

	kl	se	z
d5b	1.463137e-16	4.301595e-09	3.401382e-08
d4b	1.390712e-01	5.234011e-02	2.657067e+00
d3b	8.908828e-02	5.935784e-02	1.500868e+00
d2b	2.878758e-01	5.380953e-02	5.349903e+00
d1b	2.716303e-01	4.705336e-02	5.772813e+00
d0	9.417483e-02	5.682976e-02	1.657139e+00

We note that day -5 now has essentially zero probability.

4.5 Selected Fits to Restricted Wilcox-Weinberg Data: Days -4 to 0

In view of the very last result, we refit the model for recognized pregnancies using only days -4 to 0, with 147 pregnancies in 582 cycles.

```

> wr40 <- sbm.fit(-4:0, wr.reextract)
> wr40
log-likelihood: -323.0228
estimates (k and k*lambda):

```

	k	d4b	d3b	d2b	d1b	d0
	0.2878762	0.1390714	0.08909005	0.2878762	0.2716306	0.09416997

The sum of the daily probabilities is

```

> sum(wr40$kl)
[1] 0.8818383

```

The standard errors are

```

> wrc40<-sbm.covkl(wr40)
> wrc40

```

	kl	se	z
d4b	0.13907143	0.05233993	2.657081
d3b	0.08909005	0.05935769	1.500902
d2b	0.28787621	0.05380923	5.349941
d1b	0.27163060	0.04705336	5.772821
d0	0.09416997	0.05682905	1.657075

5 Pooled-Data Estimates

The final analyses pooled the Barrett-Marshall and Wilcox-Weinberg datasets.

5.1 The Interface Functions

The interface functions are very simple: they call the interfaces for the Barrett-Marshall and the Wilcox-Weinberg datasets and concatenate the results. Here's the 'unrestricted' interface, including all pregnancies:

```
> both.x
function(days)
{
# Concatenate Barrett-Marshall and Wilcox-Weinberg datasets
# GR 21-June-96
#
      bm <- bm.extract(days)
      ww <- w.extract(days)
      y <- c(bm$y, ww$y)
      X <- rbind(bm$X, ww$X)
      w <- rep(1, length(y))
      list(y = y, X = X, w = w)
}
```

And here's the 'restricted' interface, including only recognized pregnancies. The only difference with the previous function is that we call `wr.extract` instead of `w.extract`.

```
> bothr.x
function(days)
{
# Concatenate Barrett-Marshall and Wilcox-Weinberg (restricted) datasets
# GR 21-June-96
#
      bm <- bm.extract(days)
      ww <- wr.extract(days)
      y <- c(bm$y, ww$y)
      X <- rbind(bm$X, ww$X)
      w <- rep(1, length(y))
      list(y = y, X = X, w = w)
}
```

We delegate the task of checking the arguments to the actual Barrett-Marshall and Wilcox-Weinberg interfaces.

5.2 Selected Fits to Pooled Data: Days -5 to 1

We consider the range -5 to 1, which consists of days that were significant in at least one of the two pooled datasets, with a total of 282 pregnancies in 1027 cycles:

```
> both51 <- sbm.fit(-5:1, both.x)
> both51
log-likelihood: -567.3985
estimates (k and k*lambda):
      k      d5b      d4b      d3b      d2b      d1b      d0
0.398938 0.044049 0.1558072 0.1868634 0.2586475 0.3144481 0.2210129
      d1a
0.04861836
```

The sum of the daily probabilities is

```
> sum(both51$k1)
[1] 1.229446
```

The standard errors are

```
> bothc51<-sbm.covkl(both51)
> bothc51
      k1      se      z
d5b 0.04404900 0.02221054 1.983247
d4b 0.15580716 0.04238391 3.676092
d3b 0.18686344 0.05152917 3.626362
d2b 0.25864751 0.04865467 5.315985
d1b 0.31444810 0.04647823 6.765493
d0 0.22101290 0.05145466 4.295294
d1a 0.04861836 0.02804603 1.733521
```

Note that all days in the range -5 to 1 have non-zero probabilities.

Next we restrict the analysis to 237 recognized pregnancies among 1027 cycles:

```
> bothr51 <- sbm.fit(-5:1, bothr.x)
> bothr51
```

```

log-likelihood: -525.362
estimates (k and k*lambda):
      k      d5b      d4b      d3b      d2b      d1b      d0
0.3095693 0.03592043 0.1363871 0.1551057 0.2768329 0.2979934 0.1233484
      d1a
0.04495887

```

The sum of the daily probabilities is

```

> sum(bothr51$k1)
[1] 1.070547

```

\end{verbatim}

The standard errors are

\begin{verbatim}

```

> bothrc51 <- sbm.covkl(bothr51)
> bothrc51

```

	kl	se	z
d5b	0.03592043	0.02028739	1.770579
d4b	0.13638706	0.04009927	3.401236
d3b	0.15510573	0.04742730	3.270389
d2b	0.27683290	0.04802432	5.764431
d1b	0.29799340	0.04574082	6.514824
d0	0.12334844	0.04401761	2.802252
d1a	0.04495887	0.02698182	1.666265

And again, all days in the range have non-zero probabilities, at least by a one-tailed normal test.

References

- Barrett, J. C. and Marshall, J. (1969). The risk of conception on different days of the menstrual cycle. *Population Studies*, 23(2):455–461.
- Royston, J. P. (1982). Basal body temperature, ovulation, and the risk of conception, with special references to the lifetimes of sperm and egg. *Biometrics*, 38(2):397–406.
- Schwartz, D., MacDonald, P. D. M., and Heuchel, V. (1980). Fecundability, coital frequency, and the viability of ova. *Population Studies*, 34(2):397–400.

- Wilcox, A. J., Weinberg, C. R., and Baird, D. D. (1995). Timing of sexual intercourse in relation to ovulation: Effects on the probability of conception, survival of the pregnancy, and sex of the baby. *The New England Journal of Medicine*, 333(23):1517–1521.
- Wilcox, A. J., Weinberg, C. R., O'Connor, J. F., Baird, D. D., Schlatterer, J. P., Canfield, R. E., Armstrong, E. G., and Nisula, B. C. (1988). Incidence of early pregnancy loss. *New England Journal of Medicine*, 319(4):189–194.

A Variance-Covariance Matrices

This appendix lists the parameter estimates and the complete variance-covariance matrices for the final five models of interest.

A.1 Barrett-Marshall -5:1

```
> print.default(bmc51)
$par:
      d5b      d4b      d3b      d2b      d1b      d0
0.03841258 0.1379178 0.2015542 0.1973683 0.3445385 0.1396421
      d1a
0.06837946

$var:
      d5b      d4b      d3b      d2b
d5b  4.904309e-04 -9.823403e-05 -3.433375e-05 -4.025644e-05
d4b -9.823403e-05  2.664382e-03 -3.988274e-05 -1.146221e-04
d3b -3.433375e-05 -3.988274e-05  3.807955e-03  7.541771e-05
d2b -4.025644e-05 -1.146221e-04  7.541771e-05  4.484146e-03
d1b -9.791180e-06  1.478050e-04  3.234486e-04  4.273083e-04
d0  -1.955338e-05 -1.038710e-04 -1.450929e-04 -1.071530e-04
d1a -1.504250e-05 -6.093146e-05 -2.718314e-05 -6.364818e-05
      d1b      d0      d1a
d5b -9.791180e-06 -1.955338e-05 -1.504250e-05
d4b  1.478050e-04 -1.038710e-04 -6.093146e-05
d3b  3.234486e-04 -1.450929e-04 -2.718314e-05
d2b  4.273083e-04 -1.071530e-04 -6.364818e-05
d1b  6.089922e-03  2.227445e-04 -2.947919e-05
d0  2.227445e-04  3.092763e-03 -1.157079e-04
d1a -2.947919e-05 -1.157079e-04  1.224176e-03

attr(, "class"):
[1] "sbmck1"
```

A.2 Wilcox-Weinberg Unrestricted -5:0

```
> print.default(wc50)
$par:
```

```

          d5b      d4b      d3b      d2b      d1b      d0
0.09935487 0.155297 0.1368277 0.2747404 0.3133121 0.3316633

```

```
$var:
```

```

          d5b      d4b      d3b      d2b
d5b  6.037045e-03 -5.174991e-04 -0.0003891571 -0.0004155559
d4b -5.174991e-04  4.258256e-03 -0.0005629296 -0.0001521099
d3b -3.891571e-04 -5.629296e-04  0.0069294575 -0.0001064522
d2b -4.155559e-04 -1.521099e-04 -0.0001064522  0.0044591392
d1b -2.070491e-04 -3.401887e-05 -0.0002020755  0.0001523673
d0  -2.033227e-05 -7.761414e-06 -0.0001441285  0.0002629561
          d1b      d0
d5b -2.070491e-04 -2.033227e-05
d4b -3.401887e-05 -7.761414e-06
d3b -2.020755e-04 -1.441285e-04
d2b  1.523673e-04  2.629561e-04
d1b  3.253847e-03  4.208587e-04
d0  4.208587e-04  7.371220e-03

```

```
attr(, "class"):
```

```
[1] "sbmck1"
```

A.3 Wilcox-Weinberg Restricted -4:0

```
> print.default(wrc40)
```

```
$par:
```

```

          d4b      d3b      d2b      d1b      d0
0.1390714 0.08909005 0.2878762 0.2716306 0.09416997

```

```
$var:
```

```

          d4b      d3b      d2b      d1b
d4b  2.739468e-03 -2.017415e-04 4.649168e-05  5.567382e-06
d3b -2.017415e-04  3.523335e-03 3.826060e-05 -3.200424e-05
d2b  4.649168e-05  3.826060e-05 2.895434e-03  1.520177e-04
d1b  5.567382e-06 -3.200424e-05 1.520177e-04  2.214019e-03
d0 -3.657092e-04 -9.379182e-04 3.793740e-05  1.881760e-05
          d0
d4b -0.0003657092
d3b -0.0009379182

```

```
d2b 0.0000379374
d1b 0.0000188176
d0 0.0032295406
```

```
attr(,"class"):
[1] "sbmck1"
```

A.4 Pooled Unrestricted -5:1

```
> print.default(bothc51)
```

```
$par:
```

```
      d5b      d4b      d3b      d2b      d1b      d0
0.044049 0.1558072 0.1868634 0.2586475 0.3144481 0.2210129
      d1a
0.04861836
```

```
$var:
```

```
      d5b      d4b      d3b      d2b
d5b 4.933082e-04 -7.021249e-05 -3.014043e-05 -3.761118e-05
d4b -7.021249e-05 1.796396e-03 -7.965537e-05 -4.683912e-05
d3b -3.014043e-05 -7.965537e-05 2.655255e-03 4.589403e-05
d2b -3.761118e-05 -4.683912e-05 4.589403e-05 2.367277e-03
d1b -1.848956e-05 1.843941e-05 3.031089e-05 1.615658e-04
d0 -1.747035e-05 -8.903995e-05 -1.450869e-04 -1.116692e-05
d1a -2.842227e-05 -6.372466e-05 -7.035876e-05 -4.230500e-05
      d1b      d0      d1a
d5b -1.848956e-05 -1.747035e-05 -2.842227e-05
d4b 1.843941e-05 -8.903995e-05 -6.372466e-05
d3b 3.031089e-05 -1.450869e-04 -7.035876e-05
d2b 1.615658e-04 -1.116692e-05 -4.230500e-05
d1b 2.160226e-03 1.513038e-04 -3.220103e-05
d0 1.513038e-04 2.647582e-03 -6.166752e-05
d1a -3.220103e-05 -6.166752e-05 7.865796e-04
```

```
attr(,"class"):
[1] "sbmck1"
```

A.5 Pooled Restricted -5:1

```
> print.default(bothrc51)
```

```
$par:
```

```
      d5b      d4b      d3b      d2b      d1b      d0
0.03592043 0.1363871 0.1551057 0.2768329 0.2979934 0.1233484
      d1a
0.04495887
```

```
$var:
```

```
      d5b      d4b      d3b      d2b
d5b  4.115783e-04 -5.634376e-05 -2.450725e-05 -8.268698e-06
d4b -5.634376e-05  1.607951e-03 -7.261106e-05  8.348400e-06
d3b -2.450725e-05 -7.261106e-05  2.249349e-03  4.048047e-05
d2b -8.268698e-06  8.348400e-06  4.048047e-05  2.306335e-03
d1b -1.594512e-06  3.096321e-05  4.471568e-05  1.668521e-04
d0  -1.993991e-05 -1.287081e-04 -2.156631e-04 -4.791343e-06
d1a -2.500620e-05 -5.421232e-05 -5.357183e-05 -8.273706e-06
      d1b      d0      d1a
d5b -1.594512e-06 -1.993991e-05 -2.500620e-05
d4b  3.096321e-05 -1.287081e-04 -5.421232e-05
d3b  4.471568e-05 -2.156631e-04 -5.357183e-05
d2b  1.668521e-04 -4.791343e-06 -8.273706e-06
d1b  2.092222e-03  4.036333e-05 -2.019349e-06
d0  4.036333e-05  1.937550e-03 -8.124707e-05
d1a -2.019349e-06 -8.124707e-05  7.280189e-04
```

```
attr(, "class"):
```

```
[1] "sbmck1"
```