APPENDIX B
COMPUTER CODE FOR THE SIMULATION PROCEDURE

This appendix describes the computer code used to implement the cost-effectiveness analysis described in the case study (Chapter 6). Specifically, this code computes QALY costs and health event counts using Equation 6-7. The computer code computes the range of plausible values for these results and for the cost-effectiveness ratio using Monte Carlo methods that repeatedly assign randomly selected quantities to the parameters on the right side of Equation 6-7 and then compute and record the resulting QALY costs and health event counts.

Appendix B1 reproduces Equation 6-7 and defines all of its component terms. Appendix B2 then describes how parts of this equation are used to compute health event counts and QALY costs for specific health event types. Appendix B3 outlines the computer code developed to execute these calculations. Finally, Appendix B-4 reproduces the code used to implement that Monte Carlo analysis, and provides a table of contents to facilitate location of procedures (referred to as "macros" in SAS).

B.1. SUMMARY OF CASE STUDY QALY COST EQUATION AND DEFINITION OF TERMS

Total QALY costs (TCost) associated with consumption of tap water during the 20-year analytical time frame are computed as

$$TCost = \sum_{i=1}^{7} \frac{1}{(1+d)^{0.5}} \times \sum_{j=1}^{20} \frac{1}{(1+d)^{j-1}} \times \sum_{k=1}^{18} \Delta r_{i,k} \times N_AtRisk_{i,k} \times Popsize_k \times AveQCost_{i,l},$$

where

$$AveQCost_{i,k} = \sum_{l=0}^{MaxAge-age(k)} \Pr Latency_{i,k,l} \times \Pr Alive_{k,l} \times \frac{QVal_{i,age(k)+l}}{(1+d)^l}$$

The indexes for these equations are defined as

i	=	The index over the 7 health event types considered in this case study;
j	=	The index over the 20 year analytical time frame;
k	=	the index over 18 age cohorts comprising the population (k=1 corresponds to ages 0 to 4, k=2 corresponds to ages 5 to 9, and so forth, through k=18, which corresponds to ages 85 and above; and
l	=	The duration of the latency period – *i.e.*, the number of years between exposure and the manifestation of resulting health effect.

Other terms in Equation 6-7 include

$\Delta r_{i,k}$	=	The incremental risk of contracting the health effect each time an individual in this cohort is "at risk" for developing the condition;
$N_AtRisk_{i,k}$	=	The average number of times each member of cohort k is at risk for the health endpoint i;
$PopSize_k$	=	The number of individuals in the population belonging to age cohort k.
$PrLatency_{i,k,l}$	=	The probability that the latency period is l years for an event of type i resulting from exposure while a member of age cohort k;
$PrAlive_{k,l}$	=	The probability that a member of cohort k will still be alive after a latency period of l years; and

$QVal_{i,age(k)+l}$ = The QALY value of a health event of type *i* for an individual whose age is *l* years greater than the mean age among members of age cohort *k*.

B.2. COMPUTATION OF HEALTH EVENT COUNTS AND QALY COSTS FOR EACH EVENT TYPE

Health event counts were computed by omitting components of Equation 6-7 that weight these events to reflect their net present value when measured in terms of QALYs. Hence, the number of events of type *i* ($NumEvents_i$) is

$$NumEvents_i = \sum_{j=1}^{20} \sum_{k=1}^{18} \Delta r_{i,k} \times N_AtRisk_{i,k} \times Popsize_k$$

QALY costs for a single event type ($Cost_i$) are computed by holding event type (*i*) constant and omitting the outermost summation in equation 6-7.

$$Cost_i = \sum_{j=1}^{20} \frac{1}{(1+d)^{j-1}} \times \sum_{k=1}^{18} \Delta r_{i,k} \times N_AtRisk_{i,k} \times Popsize_k \times \sum_{l=0}^{MaxAge-age(k)} \Pr Latency_{i,k,l} \times \Pr Alive_{k,l} \times \frac{QVal_{i,age(k)+l}}{(1+d)^l}$$

B.3. PROGRAM EXECUTION OUTLINE

A computer program written in SAS version 6.12 for IBM compatible PCs (SAS, 1990) implements the Monte Carlo algorithm used to repeatedly compute the outer-most three summations of Equation 6-7 (*i.e.*, the first equation that appears in Appendix B1). The code for

the simulation appears in Appendix B4. The following table outlines the general execution of this code. Procedure names are indented below the names of those procedures by which they are called.

Table B-1
Outline of Code Execution

Macro (Procedure)	See Page	Comment
SIMULATE	B-15	Top level procedure
INIT_DB	B-15	Initializes dataset RESULTS.SD2 that holds the Monte Carlo results
ONE_ITER	B-16	Executed LOOPNUM (1,000) times; computes event counts and QALY costs
R_ASSIGN	B-16	Randomly assigns values to uncertain parameters. Values stored in ASSIGN.SD2.
MAINCALC(1)	B-16	Calculates event counts and QALY costs for the baseline technology. Uses parameters in BENEFITS.SD2. Stores calculations and uncertain parameter values in RESULTS.SD2.
MAINCALC(2)	B-16	Calculates event counts and QALY costs for the supplemental technology. Uses parameters in BENEFITS.SD2. Stores calculations and uncertain parameter values in RESULTS.SD2.
SUMMARY	B-24	Summarizes Monte Carlo results. Creates BOTH.SD2, which contains a single record for each set of random parameter values; this record has results for both the baseline and supplemental technologies.

The program uses the following data sets:

- RESULTS.SD2 – Contains 2 records for each set of randomly selected parameter values. The first contains the parameter values, event counts, and QALY costs for the baseline technology; the second contains the same information for the supplemental technology.

- BOTH.SD2 – Contains a single entry for each set of randomly drawn parameter values. This data set is created by combining each pair of corresponding records in RESULTS.SD2.

- TECH1.SD2 and TECH2.SD2 – Temporary files used in the process of creating BOTH.SD2 from RESULTS.SD2.

- ASSIGN.SD2 – Temporarily stores all randomly generated parameter values for a single iteration of the Monte Carlo simulation (see procedure R_ASSIGN).

- BENEFITS.SD2 – Fixed parameters and results for computation of QALY costs and event counts for each age cohort. The "*x*" character that appears in field names, such as "NAR_*x*" and "QPE_*x*", refers to a string that corresponds to 1 of the 7 health effects. These are: CaI (cancer illness), CaD (Cancer death), Rep (Reproductive Toxicity), Dev (Developmental Toxicity), CIm (Mild Crypto illness), CIS (Moderate to severe Crypto illness), and CD (Crypto death). Parameters stored include

    - Agegroup – A label indicating the age range of the cohort represented by each record;

    - Tot_TWIN – Total tapwater consumption per day in L/kg-day

    - UnhtTWIn – Total unheated tapwater consumption in L/day

    - risk_*x* – The value of $Dr_{i,k}$ for health endpoint *x*.

    - N – Fraction of population in this age group

    - NAR_*x* – $N_AtRisk_{i,k}$ for health endpoint *x*.

    - QPE_*x*d – $QAveCost_{i,k}$ for discount rate d for health endpoint *x*. There are entries for d = 3, 4, 5, 6, and 7 percent.

    - TotQ*x* – Total QALYs for age cohort k for health endpoint *x*. Equals the following equation:

$$\Delta r_{i,k} \times N_AtRisk_{i,k} \times Popsize_k \times AveQCost_{i,k}$$

- SUMBEN.SD2 – Temporary data set that contains the event counts and QALY totals computed by summing over all 18 records in BENEFITS.SD2.

Procedure MAINCALC implements much of the computation, calculating the number of health events and the corresponding QALY cost attributable to a single year of exposure ($YCost_i$). Specifically, this routine computes

$$YCost_i = \sum_{k=1}^{18} \Delta r_{i,k} \times N_AtRisk_{i,k} \times Popsize_k \times AveQCost_{i,k}$$

The MAINCALC routine carries out computations described below. The underlined text refers to a comment in the program just before the code that executes that task. In reviewing the code, note that variables with names that start with an ampersand ("&") are "macro" variables that are defined within the program (all macro variables are declared near the beginning of the program). Other variables refer to fields in the BENEFITS.SD2 dataset. Each record in this dataset corresponds to a single age cohort, and there are therefore 18 records in this dataset.

- <u>Crypto Risks</u>: Computes the number of health events experienced by a single age cohort due to a single year of drinking water consumption. This value is equal to $\Delta r_{i,k} \times N_AtRisk_{i,k} \times Popsize_k$. Health endpoints include mild *Cryptosporidium* illness, moderate to severe *Cryptosporidium* illness, and death due *Cryptosporidium* infection.

- <u>DBP Risks</u>: Computes the corresponding event counts for DBP-induced illness (cancer illness, cancer death, developmental toxicity, and reproductive toxicity).

- <u>NPV QALYs per health event</u>: The event counts described in the preceding two bullets are multiplied by the $QAveCost_{i,k}$. The MAINCALC routine has been written to use any discount rate between 3% and 7% by interpolating between values corresponding integral discount rates that are stored as parameters in BENEFITS.SD2.

- Sum event counts and QALY costs over age groups: Totals are computed over all 18 age cohorts for both QALY costs and

Procedure SUMMARY creates the dataset BOTH.SD2 (described above). In the process, it sums event counts and QALY costs over all 20 years of the analytic time frame; it also multiplies the QALY costs by a leading term of $\frac{1}{(1+d)^{0.5}}$. Finally, SUMMARY scales the results to reflect the size of the population (it multiplies by & PopSize).

B.4. SIMULATION CODE

Function	Page	Description
UniDist	B-13	Generates uniform random value
LUniDist	B-13	Generates log uniform random value
NormDist	B-13	Generates normal random value
PNormal	B-14	Generates non-negative random value
LNDist	B-14	Generates lognormal random value
LNDistT	B-14	Generates truncated lognormal random value
TrDist	B-14	Generates triangular distributed random value
BetaDist	B-15	Generates beta distributed random value
Simulate	B-15	Top level procedure
Init_DB	B-15	Initializes dataset RESULTS.SD2 that holds the Monte Carlo results
One_Iter	B-16	Executed LOOPNUM (1,000) times; computes event counts and QALY costs
R_Assign	B-16	Randomly assigns values to uncertain parameters. Values stored in ASSIGN.SD2
MainCalc	B-21	Calculates event counts and QALY costs. Uses parameters in BENEFITS.SD2. Stores calculations and uncertain parameter values in RESULTS.SD2.
Summary	B-24	Summarizes Monte Carlo results. Creates BOTH.SD2, which contains a single record for each set of random parameter values; this record has results for both the baseline and supplemental technologies.

```
*==============================================================================;
* CE.sas                                                                       ;
*                                                                              ;
* Computes                                                                     ;
*                                                                              ;
*     Incremental net present value of technology costs                        ;
*                                                                              ;
*     NPV of QALYs lost due to drinking water-related illnesses                ;
*         For technology 1                                                     ;
*         For technology 2                                                     ;
*         and the difference between tech 1 and tech 2                         ;
*                                                                              ;
*     Natural unit counts of cases of each health endpoint                     ;
*                                                                              ;
* Health endpoints considered are:                                             ;
*     (CaI) Cancer illness                                                     ;
*     (CaD) Cancer death                                                       ;
*     (Rep) Repro dysfunction                                                  ;
*     (Dev) Developmental disorders                                            ;
*     (CIm) Cryptosporidium mild illness                                       ;
*     (CIs) Cryptosporidium moderate to severe illness                         ;
*     (CD)  Cryptosporidium death                                              ;
*******************************************************************************;

options nonotes nosource nosource2 nosymbolgen nomlogic nomrecall nomprint;

* Location of file containing info for calculation of event counts and NPV QALY cost.;
* See macro maincalc;
    libname datapath
        'g:\projects\773214\02010000\analysis\data';


*===================================
Define constants.
*===================================;

* Note -- Before running, make the following assignments:;
*     &AIDS =  1 (AIDS population) or 0 (general population);
*     &SupHF = 1 (Home filters)    or 0 (ozone);
*     Comment out appropriate Popsize value;
*     Comment out appropriate tech costs;
*     Comment out appropriate removal efficiency values;

* Runtime constants;
    %let seed        = 0;               *Seed to be used in random number generator;
    %let loopnum     = 5   ;            *Number of uncertainty loops to execute;
    %let AIDS        = 1;               *1 for AIDS, 0 for general population;
    %let SupHF       = 1;               *1 for home filters, 0 for ozone;

* Population size;
*   %let PopSize     = 4.6e5;           *General population;
    %let PopSize     = 429;             *AIDS subpopulation;

* Plant life;
    %let plantL      = 20;              *In years;

* Discount rate;
    %let drate_l     = 0.05;            *lower bound;
    %let drate_u     = 0.05;            *upper bound;
    %let drate       = 0.050;           *Place holder;

* Per event QALY value multipliers;
    %let QCaIm_l    = 0.50; %let QCaIm_u     = 2.00; * Cancer illness;
    %let QCaDm_l    = 0.50; %let QCaDm_u     = 2.00; * Cancer death;
    %let QDevm_l    = 0.50; %let QDevm_u     = 2.00; * Developmental tox;
    %let QRepm_l    = 0.50; %let QRepm_u     = 2.00; * Reproductive tox;
    %let QCImm_l    = 0.50; %let QCImm_u     = 2.00; * Mild Crypto illness;
    %let QCIsm_l    = 0.50; %let QCIsm_u     = 2.00; * Moderate to severe Crypto illness;
    %let QCDm_l     = 0.50; %let QCDm_u      = 2.00; * Crypto death;
    %let QCaIm      = 1.0;             * Cancer illness;
    %let QCaDm      = 1.0;             * Cancer death;
    %let QDevm      = 1.0;             * Developmental tox;
    %let QRepm      = 1.0;             * Reproductive tox;
```

```
    %let QCImm      = 1.0;              * Mild Crypto illness;
    %let QCIsm      = 1.0;              * Moderate to severe Crypto illness;
    %let QCDm       = 1.0;              * Crypto death;

* Technology implementation costs -- ozone supplemental technology;
* General population.
* AIDS costs calculated by multiplying system costs by ratio of ;
* AIDS pop size to total pop size (429/460000);
* Use if &AIDS = 0 and &SupHF=0;
*   %let T1_Cap     = 7.82e7;           *Tech 1 General Population capital costs;
*   %let T1_Op      = 7.30e6;           *Tech 1 General Population annual operating costs;
*   %let T2_Cap     = 8.30e7;           *Tech 2 General Population capital costs;
*   %let T2_Op      = 7.70e6;           *Tech 2 General Population annual operating costs;

* Technology implementation costs -- ozone technology;
* AIDS costs calculated by multiplying Gen pop system costs by ratio of ;
* AIDS pop size to total pop size (429/460000);
* Use if &AIDS=1 and &SupHF=1;
*   %let T1_Cap     = 72930;            *Tech 1 AIDS Population capital costs;
*   %let T1_Op      = 6808;             *Tech 1 AIDS Population annual operating costs;
*   %let T2_Cap     = 77406;            *Tech 2 AIDS Population capital costs;
*   %let T2_Op      = 7181;             *Tech 2 AIDS Population annual operating costs;

* Technology implementation costs -- home filter supplemental technology;
* Assumes cap costs of 747.50 per person and op costs of 126.50 per person;
* Use if &SupHF=1;
    %let T1_Cap     = 0.00  ;           *Tech 1 AIDS Population capital costs;
    %let T1_Op      = 0.00  ;           *Tech 1 AIDS Population annual operating costs;
    %let T2_Cap     = 320678;           *Tech 2 AIDS Population capital costs;
    %let T2_Op      =  54269;           *Tech 2 AIDS Population annual operating costs;

* Cryptosporidium source water conc.;
    %let cc_am1     = 156.8;            *mean - distribution 1;
    %let cc_asd1    =  41.3;            *std dev - distribution 1;
    %let cc_am2     =  68.5;            *mean - distribution 2;
    %let cc_asd2    =  12.7;            *std dev - distribution 2;
    %let cc_pr1     =  0.50;            *Prob that dist 1 is used;
    %let cc         = 1;                *Place holder;

* Removal efficiency - baseline and ozone;
* Use if &SupHF=0;
    %let CR1_l      = 2.0;              *Tech 1 -- Log removal lower bound;
    %let CR1_m      = 2.0;              *Tech 1 -- Log removal mode;
    %let CR1_u      = 2.0;              *Tech 1 -- Log removal upper bound;
    %let CR1        = 2.0;              *Tech 1 -- log removal place holder;
    %let CR2_l      = 0.0;              *Tech 2 -- Incremental log removal lower bound;
    %let CR2_m      = 0.5;              *Tech 2 -- Incremental log removal mode;
    %let CR2_u      = 1.5;              *Tech 2 -- Incremental log removal upper bound;
    %let CR2        = 0.5;              *Tech 2 -- log removal place holder;

* Removal efficiency - home filters;
* Use if &SupHF=1;
    %let CR2_l      = 10.0;             *Tech 2 -- Incremental log removal lower bound;
    %let CR2_m      = 10.1;             *Tech 1 -- Incremental log removal mode;
    %let CR2_u      = 10.2;             *Tech 1 -- Incremental log removal upper bound;
    %let CR2        = 10.1;             *Tech 2 -- log removal place holder;


* Fraction of tap water that is heated -- uncertainty multiplier;
    %let UnHtM_l    = 0.8;              *Lower bound;
    %let UnHtM_u    = 1.2;              *Upper bound;
    %let UnHtM      = 1;                *Place holder;

* Cryptosporidium infectivity parameter;
    %let Cinf_gmG   = 0.0042;           *GM -- General population;
    %let Cinf_gmA   = 0.0126;           *GM -- AIDS;
    %let Cinf_gsd   = 1.59;             *GSD;
    %let Cinf       = 0.0042;           *Place holder;

* Cryptosporidium -- conditional probability of mild illness given infection;
    %let CImp_gm    = 0.40;             *GM -- General population;
    %let CImp_gsd   = 1.41;             *GSD -- General population;
    %let CImp_l     = 0.80;             *low bound -- AIDS;
```

```
    %let CImp_u     = 1.00;             *upper bound -- AIDS;
    %let CImp_m     = 0.95;             *mode -- AIDS;
    %let CImp       = 1;                *Place holder;

* Cryptosporidium -- conditional probability of moderate to severe illness given
* mild illness;
    %let CIsp_gm    = 0.15;             *GM -- General population;
    %let CIsp_gsd   = 1.41;             *GSD -- General population;
    %let CIsp_l     = 0.80;             *low bound -- AIDS;
    %let CIsp_u     = 1.00;             *upper bound -- AIDS;
    %let CIsp_m     = 0.95;             *mode -- AIDS;
    %let CIsp       = 1;                *Place holder;

* Cryptosporidium -- conditional probability of death given moderate to severe illness;
    %let CDp_aG     = 9;                * alpha for general population;
    %let CDp_bG     = 402993;           * beta for general population;
    %let CDp_aA     = 47;               * alpha for AIDS subpopulation;
    %let CDp_bA     = 817;              * beta for AIDS subpopulation;
    %let CDp        = 2E-5;             *Place holder;

* Cancer illness -- slope factor (L/kg-d)^-1;
    %let SF1Ca       = 1;               *Place holder -- SF for baseline tech;
    %let SF2Ca       = 2;               *Place holder -- SF for baseline + sup tech;

* Cancer death -- conditional probability of death given cancer illness;
    %let CaDp_gm     = 0.35;            *GM;
    %let CADp_gsd    = 1;               *GSD;
    %let CADp        = 1;               *place holder;

* Reproductive tox -- slope factor (L/kg-d)^-1;
    %let SF1Re        = 1;              *Place holder -- SF for baseline tech;
    %let SF2Re        = 2;              *Place holder -- SF for baseline + sup tech;

* Developmental tox -- slope factor (L/kg-d)^-1;
    %let SF1De        = 1;              *Place holder -- SF for baseline tech;
    %let SF2De        = 2;              *Place holder -- SF for baseline + sup tech;

* Cancer slope factor distribution parameters for identified carcinogenic agents;
* Distributions are assumed to be lognormal with specified GM and GSD statistics;
* SF is per unit mass -- i.e., (mg/kg-d)^-1);
    %let SCmGM1 = 5.70e-3; %let SCmGSD1 =  4.27;  * CHBrCl2;
    %let SCmGM2 = 7.20e-4; %let SCmGSD2 = 18.05;  * CHBr2Cl;
    %let SCmGM3 = 3.40e-4; %let SCmGSD3 =  6.77;  * CHBr3;
    %let SCmGM4 = 4.10e-2; %let SCmGSD4 =  2.02;  * CH;
    %let SCmGM5 = 1.40e-3; %let SCmGSD5 = 13.40;  * DCA;
    %let SCmGM6 = 4.90e-2; %let SCmGSD6 =  1.39;  * TCA;
    %let SCmGM7 = 3.20e-1; %let SCmGSD7 =  1.30;  * Bromate;
    %let SCmGM8 = 0.00   ; %let SCmGSD8 =  1.00;  * DBA;
    %let SCmGM9 = 0.00   ; %let SCmGSD9 =  1.00;  * BCA;
    %let SCmGM10= 0.00   ; %let SCmGSD10=  1.00;  * MBA;
    %let SCmGM11= 0.00   ; %let SCmGSD11=  1.00;  * DCAN;
    %let SCmGM12= 0.00   ; %let SCmGSD12=  1.00;  * TCAN;
    %let SCmGM13= 0.00   ; %let SCmGSD13=  1.00;  * BCAN;
    %let SCmGM14= 0.00   ; %let SCmGSD14=  1.00;  * DBAN;

* Repro slope factor distribution parameters for identified reproductive toxins;
* Distributions are assumed to be lognormal with specified GM and GSD statistics;
* SF is per unit mass -- i.e., (mg/kg-d)^-1);
    %let SRmGM1 = 0.00   ; %let SRmGSD1 =  1.00;  * CHBrCl2;
    %let SRmGM2 = 0.00   ; %let SRmGSD2 =  1.00;  * CHBr2Cl;
    %let SRmGM3 = 0.00   ; %let SRmGSD3 =  1.00;  * CHBr3;
    %let SRmGM4 = 0.00   ; %let SRmGSD4 =  1.00;  * CH;
    %let SRmGM5 = 2.50e-2; %let SRmGSD5 =  1.70;  * DCA;
    %let SRmGM6 = 0.00   ; %let SRmGSD6 =  1.00;  * TCA;
    %let SRmGM7 = 0.00   ; %let SRmGSD7 =  1.00;  * Bromate;
    %let SRmGM8 = 2.50e-2; %let SRmGSD8 =  1.70;  * DBA;
    %let SRmGM9 = 2.50e-2; %let SRmGSD9 =  1.70;  * BCA;
    %let SRmGM10= 0.00   ; %let SRmGSD10=  1.00;  * MBA;
    %let SRmGM11= 0.00   ; %let SRmGSD11=  1.00;  * DCAN;
    %let SRmGM12= 0.00   ; %let SRmGSD12=  1.00;  * TCAN;
    %let SRmGM13= 0.00   ; %let SRmGSD13=  1.00;  * BCAN;
    %let SRmGM14= 0.00   ; %let SRmGSD14=  1.00;  * DBAN;
```

```
* Developmental slope factor distribution parameters for identified developmental toxins;
* Distributions are assumed to be lognormal with specified GM and GSD statistics;
* SF is per unit mass -- i.e., (mg/kg-d)^-1);
    %let SDmGM1 = 0.00   ; %let SDmGSD1 =  1.00;  * CHBrCl2;
    %let SDmGM2 = 0.00   ; %let SDmGSD2 =  1.00;  * CHBr2Cl;
    %let SDmGM3 = 0.00   ; %let SDmGSD3 =  1.00;  * CHBr3;
    %let SDmGM4 = 0.00   ; %let SDmGSD4 =  1.00;  * CH;
    %let SDmGM5 = 8.60e-3; %let SDmGSD5 =  1.29;  * DCA;
    %let SDmGM6 = 2.00e-2; %let SDmGSD6 =  1.28;  * TCA;
    %let SDmGM7 = 0.00   ; %let SDmGSD7 =  1.00;  * Bromate;
    %let SDmGM8 = 8.60e-3; %let SDmGSD8 =  1.29;  * DBA;
    %let SDmGM9 = 8.60e-3; %let SDmGSD9 =  1.29;  * BCA;
    %let SDmGM10= 8.40e-3; %let SDmGSD10=  1.84;  * MBA;
    %let SDmGM11= 5.40e-3; %let SDmGSD11=  1.94;  * DCAN;
    %let SDmGM12= 2.10e-3; %let SDmGSD12=  1.34;  * TCAN;
    %let SDmGM13= 1.60e-1; %let SDmGSD13=  1.28;  * BCAN;
    %let SDmGM14= 2.10e-1; %let SDmGSD14=  1.34;  * DBAN;

* DBP Concentration distributions for the water treated using the baseline technology;
* Distributions are assumed to be normal with specified arithmetic mean (AM) and
* arithmetic standard deviation (ASD) statistics;
* Concentration is in ug/L;
    %let C1_AM1 = 24.4   ; %let C1_ASD1 = 1.52   ;  * CHBrCl2;
    %let C1_AM2 = 10.2   ; %let C1_ASD2 = 8.51e-1;  * CHBr2Cl;
    %let C1_AM3 =  0.35  ; %let C1_ASD3 = 2.55e-1;  * CHBr3;
    %let C1_AM4 =  4.2   ; %let C1_ASD4 = 3.34e-1;  * CH;
    %let C1_AM5 = 30.85  ; %let C1_ASD5 = 1.49   ;  * DCA;
    %let C1_AM6 = 20.1   ; %let C1_ASD6 = 9.42e-1;  * TCA;
    %let C1_AM7 =  0.00  ; %let C1_ASD7 = 1.00   ;  * Bromate;
    %let C1_AM8 =  1.5   ; %let C1_ASD8 = 1.22e-1;  * DBA;
    %let C1_AM9 =  8.5   ; %let C1_ASD9 = 9.12e-2;  * BCA;
    %let C1_AM10=  0.29  ; %let C1_ASD10= 2.74e-2;  * MBA;
    %let C1_AM11=  3.5   ; %let C1_ASD11= 4.56e-1;  * DCAN;
    %let C1_AM12=  0.2   ; %let C1_ASD12= 7.60e-2;  * TCAN;
    %let C1_AM13=  1.9   ; %let C1_ASD13= 2.43e-1;  * BCAN;
    %let C1_AM14=  0.15  ; %let C1_ASD14= 7.30e-2;  * DBAN;
    %let C1_AMCa= 83     ; %let C1_ASDCa= 1.28e+1;  * Unspecified carcinogens;
    %let C1_AMDe= 47     ; %let C1_ASDDe= 6.99   ;  * Unspecified developmental toxins;
    %let C1_AMRe= 47     ; %let C1_ASDRe= 6.99   ;  * Unspecified reproductive toxins;


* DBP Concentration distributions for the water treated using the baseline technology;
* Distributions are assumed to be normal with specified arithmetic mean (AM) and
* arithmetic standard deviation (ASD) statistics;
* Concentration is in ug/L;
    %let C2_AM1 = 21.1   ; %let C2_ASD1 = 1.52e-1;  * CHBrCl2;
    %let C2_AM2 = 13.0   ; %let C2_ASD2 = 4.86e-1;  * CHBr2Cl;
    %let C2_AM3 =  1.5   ; %let C2_ASD3 = 2.13e-1;  * CHBr3;
    %let C2_AM4 =  5.8   ; %let C2_ASD4 = 5.78e-1;  * CH;
    %let C2_AM5 = 19.3   ; %let C2_ASD5 = 7.90e-1;  * DCA;
    %let C2_AM6 = 10     ; %let C2_ASD6 = 6.99e-1;  * TCA;
    %let C2_AM7 =  4     ; %let C2_ASD7 = 3.65e-1;  * Bromate;
    %let C2_AM8 =  1.98  ; %let C2_ASD8 = 1.40e-1;  * DBA;
    %let C2_AM9 =  6.7   ; %let C2_ASD9 = 1.22e-1;  * BCA;
    %let C2_AM10=  0.28  ; %let C2_ASD10= 3.65e-2;  * MBA;
    %let C2_AM11=  2.6   ; %let C2_ASD11= 2.43e-1;  * DCAN;
    %let C2_AM12=  0.05  ; %let C2_ASD12= 0.00   ;  * TCAN;
    %let C2_AM13=  1.65  ; %let C2_ASD13= 1.25e-1;  * BCAN;
    %let C2_AM14=  0.55  ; %let C2_ASD14= 1.43e-1;  * DBAN;
    %let C2_AMCa= 68     ; %let C2_ASDCa= 1.19e+1;  * Unspecified carcinogens;
    %let C2_AMDe= 39     ; %let C2_ASDDe= 6.99   ;  * Unspecified developmental toxins;
    %let C2_AMRe= 39     ; %let C2_ASDRe= 6.99   ;  * Unspecified reproductive toxins;

*==============================================================================;
* Random number macros.                                                        ;
*==============================================================================;

* UniDist;
* Parameters;
*    lower   -- lower bound of uniform;
*    upper   -- upper bound of uniform;

     %macro UniDist(lower, upper);
```

```
        value = &lower + uniform(&seed)*(&upper - &lower);
    %mend UniDist;

* LUniDist;
* Parameters;
*    lower   -- lower bound of log uniform;
*    upper   -- upper bound of log uniform;
*
* Returns value whose log is uniformly distributed between log(lower) and;
* log(upper);

    %macro LUniDist(lower, upper);
        Llower = log(&lower);
        Lupper = log(&upper);
        %UniDist(Llower,Lupper);
        value = exp(value);
    %mend LUniDist;


* Normdist;
* Parameters;
*    mean     -- normal distribution arithmetic mean;
*    std      -- normal distribution standard deviation;

    %macro NormDist(mean, std);
        value = &mean + &std*normal(&seed);
    %mend NormDist;

* pNormal;
* Parameters;
*    mean     -- normal distribution arithmetic mean;
*    std      -- normal distribution standard deviation;
*
* Repeatedly draws values from a normal distribution until getting a non-negative value;

    %macro pnormal(mean, std);
       value=-1;
       do until(value ge 0);
          %NormDist(&mean, &std);
       end;
    %mend pnormal;

* LnDist;
* Parameters;
*    gm     -- geometric mean;
*    gsd    -- geometric standard deviation;

   %macro LnDist(gm, gsd);
       mean=log(&gm);
       std=log(&gsd);
       %NormDist(mean, std);
       value=exp(value);
   %mend LnDist;

* LnDistT;
* Parameters;
*    gm       -- geometric mean;
*    gsd      -- geometric standard deviation;
*    LowBound -- lower truncation bound;
*    HiBound  -- upper truncation bound;
*
* LnDistT randomly selects a value between LowBound and Hibound from the lognormal;
* defined by gm and gsd.  If LnDist(&gm,&gsd) returns the missing value (e.g., because the;
* GM is zero), then LnDistT returns 0.  Otherwise, a random value is repeatedly selected until;
* one is selected that falls within the bounds.;

   %macro LnDistT(gm, gsd, LowBound, HiBound);
       if (&gm le 0 or &gsd le 0) then value = 0; else do;
           mean=log(&gm);
           std =log(&gsd);
           do until(value ge &LowBound and value le &HiBound);
               %NormDist(mean, std);
               value=exp(value);
```

```
            end;
        end;
    %mend LnDistT;

* TrDist;
* Parameters;
*    LowBound -- Distribution lower bound;
*    Mode     -- Distribution mode;
*    HiBound  -- Distribution upper bound;

    %macro TrDist(LowBound, Mode, HiBound);
        prob=uniform(&seed);
        height=2/(&HiBound-&LowBound);

        prob1=prob;
        slope1=height/(&Mode-&LowBound);
        val1=&LowBound+(2*prob1/slope1)**0.5;

        prob2=1-prob;
        slope2=height/(&HiBound-&Mode);
        val2=&HiBound-(2*prob2/slope2)**0.5;

        if prob le (&Mode-&LowBound)/(&HiBound-&LowBound) then value = val1;
        if prob gt (&Mode-&LowBound)/(&HiBound-&LowBound) then value = val2;
        if &LowBound eq &HiBound then value = &Mode;
    %Mend TrDist;

* BetaDist;
* Parameters;
*    alpha -- first shape parameter;
*    beta  -- second shape parameter;

    %macro BetaDist(alpha,beta);
        prob=uniform(&seed);
        value = betainv(prob,&alpha,&beta)
    %mend betadist;


*==============================================================================;
* simulate                                                                     ;
* Runs outer simulation loop by calling iter &loopnum times.                   ;
*==============================================================================;

%macro simulate;

    * Initialize results database;
        %init_db;

    * Execute iteration loop &loopnum times;
        %do iter=1 %to &loopnum;
                %one_iter;
        %end;

    * Compute the incremental QALYs and NEVs (numbers of events) and summarize the results;
        %summary;

%mend simulate;

*==============================================================================;
* init_db                                                                      ;
* Creates results dataset to hold simulation output.  Sets labels.             ;
*==============================================================================;

%macro init_db;

    * Set up dummy record and establish labels;
    data results; set dummy(drop=foo);
        TechFlag = 1;         label TechFlag ='Technology';
        RunNum   = 1;         label RunNum =  'Simulation iteration';
        cc       = &cc;       label cc =      'Starting crypto conc (org/L)';
        CR1      = &CR1;      label CR1 =     'Base Tech log removal';
        CR2      = &CR2;      label CR2 =     'Base + Sup Tech log removal';
        UnHtM    = &UnHtM;    label UnHtM =   'Fraction tap water not htd -- mltplr';
```

```
        Cinf     = &Cinf;     label Cinf =    'Crypto infectivity parameter';
        CImP     = &CImP;     label CImP =    'Cond prob mild illness if infected';
        CIsP     = &CIsP;     label CIsP =    'Cond prob mod-sev ill if mildly ill';
        CDP      = &CDP;      label CDP =     'Cond prob die if mod to sev ill';
        SF1Ca    = &SF1Ca;    label SF1Ca =   'Tech 1 Cancer SF L/kg-day^-1';
        SF2Ca    = &SF2Ca;    label SF2Ca =   'Tech 2 Cancer SF L/kg-day^-1';
        SF1Re    = &SF1Re;    label SF1Re =   'Tech 1 Repro SF L/kg-day^-1';
        SF2Re    = &SF2Re;    label SF2Re =   'Tech 2 Repro SF L/kg-day^-1';
        SF1De    = &SF1De;    label SF1De =   'Tech 1 Devel SF L/kg-day^-1';
        SF2De    = &SF2De;    label SF2De =   'Tech 2 Devel SF L/kg-day^-1';
        CaDP     = &CaDP;     lable CaDP  =   'Cond prob death given cancer';
        drate    = &drate;    label drate =   'Discount rate';
        QCaIM    = &QCaIM;    label QCaIM =   'Cancer Illness QALY mult';
        QCaDM    = &QCaDM;    label QCaDM =   'Cancer Death QALY mult';
        QDevM    = &QDevM;    label QDevM =   'Dev Tox QALY mult';
        QRepM    = &QRepM;    label QRepM =   'Rep Tox QALY mult';
        QCImM    = &QCImM;    label QCImM =   'Crypto mild ill QALY mult';
        QCIsM    = &QCIsM;    label QCIsM =   'Crypto mod to sev ill QALY mult';
        QCDM     = &QCDM;     label QCDM  =   'Crypto death QALY mult';
        TotQCaI  = 0;         label TotQCaI = 'QALYs: Cancer Illness';
        TotQCaD  = 0;         label TotQCaD = 'QALYs: Cancer Death';
        TotQDev  = 0;         label TotQDev = 'QALYs: Developmental';
        TotQRep  = 0;         label TotQRep = 'QALYs: Reproductive';
        TotQCIm  = 0;         label TotQCIm = 'QALYs: Crypto - Mild Illness';
        TotQCIs  = 0;         label TotQCIs = 'QALYs: Crypto - Mod to Sev Illness';
        TotQCD   = 0;         label TotQCD  = 'QALYs: Crypto - Death';
        TotQ     = 0;         label TotQ    = 'QALYs: All Health Effects';
        NEV_CaI  = 0;         label NEV_CaI = 'Num Events: Cancer Illness';
        NEV_CaD  = 0;         label NEV_CaD = 'Num Events: Cancer Death';
        NEV_Dev  = 0;         label NEV_Dev = 'Num Events: Developmental';
        NEV_Rep  = 0;         label NEV_Rep = 'Num Events: Reproductive';
        NEV_CIm  = 0;         label NEV_CIm = 'Num Events: Crypto - Mild Illness';
        NEV_CIs  = 0;         label NEV_CIs = 'Num Events: Crypto - Mod to Sev Illness';
        NEV_CD   = 0;         label NEV_CD  = 'Num Events: Crypto - Death';
        IncTech  = 0;         label IncTech = 'Inc. Tech Cost';
        CE       = 0;         label CE      = 'Dollar per QALY';
    run;

    * Eliminate blank record.;
    data results; set results;
        if 1=2;
    run;


%mend init_db;


*==============================================================================;
* one_iter                                                                     ;
* Runs outer simulation loop by calling iter &loopnum times.                   ;
*==============================================================================;

%macro one_iter;

    * Assign random values to uncertain parameters;
       %R_Assign;

    * Calculate baseline technology health event counts and NPV QALYs;
        %maincalc(1);

    * Calculate baseline + supplemental technology health event counts and NPV QALYs;
        %maincalc(2);

%mend one_iter;


*==============================================================================;
* R_Assign                                                                     ;
* Assigns random values to uncertain parameters.                               ;
*==============================================================================;


%macro R_Assign;
```

```
* Store random values in the Assign dataset as they are calculated.  After all;
* calculations, these values are transfered to macro variables for use in the ;
* maincalc routine.;
data Assign; set dummy;

***********************;
* Economic quantities  ;
***********************;

* Annual discount rate;
    %UniDist(&drate_l,&drate_u);
    drate=value;
    call symput('drate',drate);

* Per event QALY cost multipliers;
    %LUniDist(&QCaIm_l,&QCaIm_u); QCaIm=value; call symput('QCaIm',QCaIm);
    %LUniDist(&QCaDm_l,&QCaDm_u); QCaDm=value; call symput('QCaDm',QCaDm);
    %LUniDist(&QDevm_l,&QDevm_u); QDevm=value; call symput('QDevm',QDevm);
    %LUniDist(&QRepm_l,&QRepm_u); QRepm=value; call symput('QRepm',QRepm);
    %LUniDist(&QCImm_l,&QCImm_u); QCImm=value; call symput('QCImm',QCImm);
    %LUniDist(&QCIsm_l,&QCIsm_u); QCIsm=value; call symput('QCIsm',QCIsm);
    %LUniDist(&QCDm_l, &QCDm_u) ; QCDm= value; call symput('QCDm', QCDm);

***********************;
* Crypto health effects;
***********************;

* Crypto concentration in source water (organisms/L);
* Use distribution 1 with probability &cc_pr1;
    %UniDist(0,1);
    if value le &cc_pr1 then do;
         %pnormal(&cc_am1,&cc_asd1);
         end;
    else do;
         %pnormal(&cc_am2,&cc_asd2);
         end;
    cc=value;
    call symput('cc',cc);

* Crypto log removal for baseline technology;
    %TrDist(&CR1_l,&CR1_m,&CR1_u);
    CR1=value;
    call symput('CR1',CR1);

* Crypto log removal for baseline + supplemental technology;
    %TrDist(&CR2_l,&CR2_m,&CR2_u);
    CR2=value;
    call symput('CR2',CR2);

* Multiplier to scale the fraction of ingested tap water that is not heated;
    %UniDist(&UnHtM_l,&UnHtM_u);
    UnHtM=value;
    call symput('UnHtM',UnHtM);

* Crypto infectivity parameter;
    if &AIDS=1 then do;
         %LnDist(&Cinf_gmA,&Cinf_gsd);
         end;
    else do;
         %LnDist(&Cinf_gmG,&Cinf_gsd);
         end;
    Cinf=value;
    call symput('Cinf',Cinf);

* Conditional probability of mild crypto illness given infection;
    if &AIDS=1 then do;
       %TrDist(&CImP_l,&CImP_m,&CImP_u);
       end;
    else do;
       %LnDistT(&CImP_gm,&CImP_gsd,0,1);
    end;
    CImP=value;
```

```
    call symput('CImP',CImP);

* Conditional probability of moderate to severe crypto illness given mild illness;
    if &AIDS=1 then do;
       %TrDist(&CIsP_l,&CIsP_m,&CIsP_u);
       end;
    else do;
       %LnDistT(&CIsP_gm,&CIsP_gsd,0,1);
    end;
    CIsP=value;
    call symput('CIsP',CIsP);

* Conditional probability of death given moderate to severe illness;
    if &AIDS=1 then do;
         %BetaDist(&CDp_aA,&CDp_bA);
         end;
    else do;
         %BetaDist(&CDp_aG,&CDp_bG);
         end;
    CDp=value;
    call symput('CDp',CDp);

********************;
* DBP health effects;
********************;

* Select DBP concentrations -- baseline technology;
    %pnormal(&C1_AM1, &C1_ASD1);  C1_1=value;
    %pnormal(&C1_AM2, &C1_ASD2);  C1_2=value;
    %pnormal(&C1_AM3, &C1_ASD3);  C1_3=value;
    %pnormal(&C1_AM4, &C1_ASD4);  C1_4=value;
    %pnormal(&C1_AM5, &C1_ASD5);  C1_5=value;
    %pnormal(&C1_AM6, &C1_ASD6);  C1_6=value;
    %pnormal(&C1_AM7, &C1_ASD7);  C1_7=value;
    %pnormal(&C1_AM8, &C1_ASD8);  C1_8=value;
    %pnormal(&C1_AM9, &C1_ASD9);  C1_9=value;
    %pnormal(&C1_AM10,&C1_ASD10); C1_10=value;
    %pnormal(&C1_AM11,&C1_ASD11); C1_11=value;
    %pnormal(&C1_AM12,&C1_ASD12); C1_12=value;
    %pnormal(&C1_AM13,&C1_ASD13); C1_13=value;
    %pnormal(&C1_AM14,&C1_ASD14); C1_14=value;
    %pnormal(&C1_AMCa,&C1_ASDCa); C1_Ca=value;
    %pnormal(&C1_AMDe,&C1_ASDDe); C1_De=value;
    %pnormal(&C1_AMRe,&C1_ASDRe); C1_Re=value;
    array c1{14} C1_1-C1_14;

* DBP health effects -- select DBP concentrations:  Baseline and sup technology;
    %pnormal(&C2_AM1, &C2_ASD1);  C2_1=value;
    %pnormal(&C2_AM2, &C2_ASD2);  C2_2=value;
    %pnormal(&C2_AM3, &C2_ASD3);  C2_3=value;
    %pnormal(&C2_AM4, &C2_ASD4);  C2_4=value;
    %pnormal(&C2_AM5, &C2_ASD5);  C2_5=value;
    %pnormal(&C2_AM6, &C2_ASD6);  C2_6=value;
    %pnormal(&C2_AM7, &C2_ASD7);  C2_7=value;
    %pnormal(&C2_AM8, &C2_ASD8);  C2_8=value;
    %pnormal(&C2_AM9, &C2_ASD9);  C2_9=value;
    %pnormal(&C2_AM10,&C2_ASD10); C2_10=value;
    %pnormal(&C2_AM11,&C2_ASD11); C2_11=value;
    %pnormal(&C2_AM12,&C2_ASD12); C2_12=value;
    %pnormal(&C2_AM13,&C2_ASD13); C2_13=value;
    %pnormal(&C2_AM14,&C2_ASD14); C2_14=value;
    %pnormal(&C2_AMCa,&C2_ASDCa); C2_Ca=value;
    %pnormal(&C2_AMDe,&C2_ASDDe); C2_De=value;
    %pnormal(&C2_AMRe,&C2_ASDRe); C2_Re=value;
    array c2{14} C2_1-C2_14;

* DBP Cancer effects -- Select cancer slope factors;
    %LnDistT(&SCmGM1, &SCmGSD1,0,1);  SFCa_1 =value;
    %LnDistT(&SCmGM2, &SCmGSD2,0,1);  SFCa_2 =value;
    %LnDistT(&SCmGM3, &SCmGSD3,0,1);  SFCa_3 =value;
    %LnDistT(&SCmGM4, &SCmGSD4,0,1);  SFCa_4 =value;
    %LnDistT(&SCmGM5, &SCmGSD5,0,1);  SFCa_5 =value;
    %LnDistT(&SCmGM6, &SCmGSD6,0,1);  SFCa_6 =value;
```

```
    %LnDistT(&SCmGM7, &SCmGSD7,0,1);  SFCa_7 =value;
    %LnDistT(&SCmGM8, &SCmGSD8,0,1);  SFCa_8 =value;
    %LnDistT(&SCmGM9, &SCmGSD9,0,1);  SFCa_9 =value;
    %LnDistT(&SCmGM10,&SCmGSD10,0,1); SFCa_10=value;
    %LnDistT(&SCmGM11,&SCmGSD11,0,1); SFCa_11=value;
    %LnDistT(&SCmGM12,&SCmGSD12,0,1); SFCa_12=value;
    %LnDistT(&SCmGM13,&SCmGSD13,0,1); SFCa_13=value;
    %LnDistT(&SCmGM14,&SCmGSD14,0,1); SFCa_14=value;
    array SFCa{14} SFCa_1-SFCa_14;

* DBP Developmental effects -- Select developmental slope factors;
    %LnDistT(&SDmGM1, &SDmGSD1,0,1);  SFDe_1=value;
    %LnDistT(&SDmGM2, &SDmGSD2,0,1);  SFDe_2=value;
    %LnDistT(&SDmGM3, &SDmGSD3,0,1);  SFDe_3=value;
    %LnDistT(&SDmGM4, &SDmGSD4,0,1);  SFDe_4=value;
    %LnDistT(&SDmGM5, &SDmGSD5,0,1);  SFDe_5=value;
    %LnDistT(&SDmGM6, &SDmGSD6,0,1);  SFDe_6=value;
    %LnDistT(&SDmGM7, &SDmGSD7,0,1);  SFDe_7=value;
    %LnDistT(&SDmGM8, &SDmGSD8,0,1);  SFDe_8=value;
    %LnDistT(&SDmGM9, &SDmGSD9,0,1);  SFDe_9=value;
    %LnDistT(&SDmGM10,&SDmGSD10,0,1); SFDe_10=value;
    %LnDistT(&SDmGM11,&SDmGSD11,0,1); SFDe_11=value;
    %LnDistT(&SDmGM12,&SDmGSD12,0,1); SFDe_12=value;
    %LnDistT(&SDmGM13,&SDmGSD13,0,1); SFDe_13=value;
    %LnDistT(&SDmGM14,&SDmGSD14,0,1); SFDe_14=value;
    array SFDe{14} SFDe_1-SFDe_14;

* DBP Reproductive effects -- Select Reproductive slope factors;
    %LnDistT(&SRmGM1, &SRmGSD1,0,1);  SFRe_1=value;
    %LnDistT(&SRmGM2, &SRmGSD2,0,1);  SFRe_2=value;
    %LnDistT(&SRmGM3, &SRmGSD3,0,1);  SFRe_3=value;
    %LnDistT(&SRmGM4, &SRmGSD4,0,1);  SFRe_4=value;
    %LnDistT(&SRmGM5, &SRmGSD5,0,1);  SFRe_5=value;
    %LnDistT(&SRmGM6, &SRmGSD6,0,1);  SFRe_6=value;
    %LnDistT(&SRmGM7, &SRmGSD7,0,1);  SFRe_7=value;
    %LnDistT(&SRmGM8, &SRmGSD8,0,1);  SFRe_8=value;
    %LnDistT(&SRmGM9, &SRmGSD9,0,1);  SFRe_9=value;
    %LnDistT(&SRmGM10,&SRmGSD10,0,1); SFRe_10=value;
    %LnDistT(&SRmGM11,&SRmGSD11,0,1); SFRe_11=value;
    %LnDistT(&SRmGM12,&SRmGSD12,0,1); SFRe_12=value;
    %LnDistT(&SRmGM13,&SRmGSD13,0,1); SFRe_13=value;
    %LnDistT(&SRmGM14,&SRmGSD14,0,1); SFRe_14=value;
    array SFRe{14} SFRe_1-SFRe_14;

 * Calculate total concentration for specified;
 *     baseline carcinogens:              CSp_1Ca;
 *     baseline + sup carcinogens:        CSp_2Ca;
 *     baseline dev tox substances:       CSp_1De;
 *     baseline + sup dev tox substances: CSp_2De;
 *     baseline rep tox substances:       CSp_1Re;
 *     baseline + sup rep tox substances: CSp_2Re;

 * Calculate total volume slope factor for specified;
 *     baseline carcinogens:              SF1Ca;
 *     baseline + sup carcinogens:        SF2Ca;
 *     baseline dev tox substances:       SF1De;
 *     baseline + sup dev tox substances: SF2De;
 *     baseline rep tox substances:       SF1Re;
 *     baseline + sup rep tox substances: SF2Re;


 * Initialize all totals to zero;
    CSp_1Ca=0; SF1Ca=0;
    CSp_2Ca=0; SF2Ca=0;
    CSp_1De=0; SF1De=0;
    CSp_2De=0; SF2De=0;
    CSp_1Re=0; SF1Re=0;
    CSp_2Re=0; SF2Re=0;

 * Loop through all {14} identified constituents.  Add to slope factor and to;
 * concentration total only if the slope factor for that disease and constituent;
 * is non-missing.  Divide by 1000 b/c slope factor is in mg/kg-day and conc;
 * is in ug/L.;
```

```
        do i=1 to dim(c1);
            if SFCa{i} ne 0  then do;
               CSp_1Ca=CSp_1Ca+c1{i}; SF1Ca=SF1Ca+c1{i}*SFCa{i}/1000;
               CSp_2Ca=CSp_2Ca+c2{i}; SF2Ca=SF2Ca+c2{i}*SFCa{i}/1000;
            end;
            if SFDe{i} ne 0  then do;
               CSp_1De=CSp_1De+c1{i}; SF1De=SF1De+c1{i}*SFDe{i}/1000;
               CSp_2De=CSp_2De+c2{i}; SF2De=SF2De+c2{i}*SFDe{i}/1000;
            end;
            if SFRe{i} ne 0  then do;
               CSp_1Re=CSp_1Re+c1{i}; SF1Re=SF1Re+c1{i}*SFRe{i}/1000;
               CSp_2Re=CSp_2Re+c2{i}; SF2Re=SF2Re+c2{i}*SFRe{i}/1000;
            end;
        end;

     * Inflate the slope factors to reflect unspecified toxins;
        SF1Ca=SF1Ca*(1+C1_Ca/CSp_1Ca);
        SF2Ca=SF2Ca*(1+C2_Ca/CSp_2Ca);
        SF1De=SF1De*(1+C1_De/CSp_1De);
        SF2De=SF2De*(1+C2_De/CSp_2De);
        SF1Re=SF1Re*(1+C1_Re/CSp_1Re);
        SF2Re=SF2Re*(1+C2_Re/CSp_2Re);

     * Divide by 70 to reflect the fact that this computation addresses ingestion for 1 year;
     * and the slope factor is applicable to the average consumption rate over a life time.;
        SF1Ca=SF1Ca/70;
        SF2Ca=SF2Ca/70;
        SF1De=SF1De/70;
        SF2De=SF2De/70;
        SF1Re=SF1Re/70;
        SF2Re=SF2Re/70;

     * Store the slope factors in macro variables to be used by macro maincalc;
        call symput('SF1Ca',SF1Ca);
        call symput('SF2Ca',SF2Ca);
        call symput('SF1De',SF1De);
        call symput('SF2De',SF2De);
        call symput('SF1Re',SF1Re);
        call symput('SF2Re',SF2Re);

     * Draw and store conditional probability of cancer fatality given cancer illness;
        %LnDist(&CaDP_gm,&CaDP_gsd);
        CaDP=value;
        call symput('CaDP',CaDP);

     * End data step for data set Assign;
        run;

%mend R_Assign;

*==============================================================================;
* maincalc                                                                     ;
* Calculates values (stored in benefits.sd2) used to compute NPV QALYs and     ;
* number of health events.                                                     ;
*==============================================================================;

* Structure of database benefits.sd2
*
*     Each record contains information quantifying NPV QALY costs and number of health
*     events for one age group (a five year range) resulting from a single year of
*     drinking water consumption.
*
*     Note that fields whose value is calculated here are marked with an asterisk.
*
*     An "_x" following the name of a field indicates that there are seven such fields,
*     one for each health effect.  The field name for each health effect is the prefix
*     shown, followed by a "_" and one of the following codes:
*
*     _CaI  -- Cancer - illness
*     _CaD  -- Cancer - death
*     _Rep  -- Reproductive tox
*     _Dev  -- Developmental tox
*     _CIm  -- Crypto - Mild illness
```

```
*     _CIs  -- Crypto - Moderate to severe illness
*     _CD   -- Crypto - Death
*
*
*     Field        Comment
*     --------     -------
*     Agegroup  -- Values 1 through 18 representing 5 year incremental age groups
*
*     Tot_TWIn  -- L/kg-day total tap water -- used for CAI, CAD, Rep, and Dev
*     UnhtTWIn  -- L/day unheated tap water -- used for crypto illness and death
*
*    *risk_x    -- Risk reflecting 1 year of drinking water consumption.
*
*     N         -- Fraction of population in this age group
*
*     NAR_x     -- (N_AtRisk) Number of potential health effects per individual.  The
*                  product of delta_r and N_AtRisk is the expected number of health
*                  effects per member of this age group.
*
*    *NEV_x     -- (N_Events) Number of events among individuals in this age group.  This
*                  value equals the product of dr, N, and NAR.
*
*     QpE_xi    -- NPV of the QALY cost per event given discount rate i.  For a
*                  specific discount rate, this value is produced by the excel spreadsheet
*                  accompanying this program.
*
*    *QpE_x     -- NPV of the QALY cost per event given the discount rate selected.  Computed
*                  from the QpEi_x via linear interpolation.
*
*    *TotQx    -- Total QALY cost for this age group.  Equals the product of NEV_x and
*                  QpE_x
;

* Parameter
*     Tech -- 1 if the baseline technology is being evaluated;
*             2 if the baseline plus the supplemental technology is being evaluated;

%macro maincalc(Tech);

    * Calculate event counts and QALY costs and store in dataset benefits;
    data benefits; set datapath.benefits;

    **************;
    * Crypto risks;
    **************;

    * fin_cc is the finished water crypto concentration;
        if (&Tech = 1) then do; fin_cc = &cc/(10**&CR1); end;
        if (&Tech = 2) then do; fin_cc = &cc/(10**(&CR1+&CR2)); end;

    * C_dose is the number of crypto ingested per 12 week (84 day) period;
        C_dose = fin_cc*UnhtTWIn*&UnHtM*12*7;

    * InfP is the probability of infection in a 12 week period;
        InfP = 1 - exp(-1*&Cinf*C_dose);

    * risk_CIm is the probability of mild crypto illness per 12 week period;
        risk_CIm = &CImp*InfP;
        NEV_CIm = risk_CIm*NAR_CIm*N;

    * risk_CIs is the probability of moderate to severe illness per 12 week period;
        risk_CIs = &CIsp*risk_CIm;
        NEV_CIs = risk_CIs*NAR_CIs*N;

    * risk_CD is the probability of death per year.  Calculate the probability of no;
    * death in a 12 week period.  Raise to 4.33 to get no death in 1 year.  Subtract from;
    * unity to get the probability that death will occur in 1 year;
        risk_CD = 1 - (1 - &CDp*risk_CIs)**4.33;
        NEV_CD =  risk_CD*NAR_CD*N;

    ***********;
    * DBP risks;
    ***********;
```

```
* risk_CaI is the probability of cancer illness;
    * Supplemental technology is NOT homefilters... Technology DOES affect cancer SF;
    if &SupHF ne 1 then do;
        if &Tech = 1 then do; risk_CaI=&SF1Ca*Tot_TWIn; end;
        if &Tech = 2 then do; risk_CaI=&SF2Ca*Tot_TWIn; end;
        end;
    * Supplemental technology is homefilters... use baseline cancer slope factor;
    * Technology does not affect cancer SF;
    else do;
        risk_CaI=&SF1Ca*Tot_TWIn;
        end;
    NEV_CaI  = risk_CaI*NAR_CaI*N;

* risk_CaD is the probability of cancer death;
    risk_CaD = risk_CaI*&CADp;
    NEV_CaD  = risk_CaD*NAR_CaD*N;

* risk_Rep is the probability of Repro tox;
    if &Tech = 1 then do; risk_Rep=&SF1Re*Tot_TWIn; end;
    if &Tech = 2 then do; risk_Rep=&SF2Re*Tot_TWIn; end;
    NEV_Rep  = risk_Rep*NAR_Rep*N;

* risk_Dev is the probability of Repro tox;
    if &Tech = 1 then do; risk_Dev=&SF1De*Tot_TWIn; end;
    if &Tech = 2 then do; risk_Dev=&SF2De*Tot_TWIn; end;
    NEV_Dev  = risk_Dev*NAR_Dev*N;

****************************;
* NPV QALYs per health event;
****************************;

* Calculate greatest integer less than discount rate (FLdrate) and smallest integer;
* greater than discount rate (CLdrate);
    FLdrate=floor(100*&drate);
    CLdrate=ceil(100*&drate);

* Define arrays holding NPV for QALYs for each discount rate between 3% and 7%;
    array QpExCaI{5} QpE_CaI3-QpE_CaI7;
    array QpExCaD{5} QpE_CaD3-QpE_CaD7;
    array QpExDev{5} QpE_Dev3-QpE_Dev7;
    array QpExRep{5} QpE_Rep3-QpE_Rep7;
    array QpExCIm{5} QpE_CIm3-QpE_CIm7;
    array QpExCIs{5} QpE_CIs3-QpE_CIs7;
    array QpExCD{5}  QpE_CD3-QpE_CD7;

* QpE_x is the QALY per event value for &drate as interpolated between FLdrate;
* and CLdrate.  Factor in the uncertainty multiplier for each per event QALY value.;
    QpE_CaI=&QCaIm*
        (QpExCaI{FLdrate-2}+(100*&drate-FLdrate)*(QpExCaI{CLdrate-2}-QpExCaI{FLdrate-2}));
    QpE_CaD=&QCaDm*
        (QpExCaD{FLdrate-2}+(100*&drate-FLdrate)*(QpExCaD{CLdrate-2}-QpExCaD{FLdrate-2}));
    QpE_Dev=&QDevm*
        (QpExDev{FLdrate-2}+(100*&drate-FLdrate)*(QpExDev{CLdrate-2}-QpExDev{FLdrate-2}));
    QpE_Rep=&QRepm*
        (QpExRep{FLdrate-2}+(100*&drate-FLdrate)*(QpExRep{CLdrate-2}-QpExRep{FLdrate-2}));
    QpE_CIm=&QCImm*
        (QpExCIm{FLdrate-2}+(100*&drate-FLdrate)*(QpExCIm{CLdrate-2}-QpExCIm{FLdrate-2}));
    QpE_CIs=&QCIsm*
        (QpExCIs{FLdrate-2}+(100*&drate-FLdrate)*(QpExCIs{CLdrate-2}-QpExCIs{FLdrate-2}));
    QpE_CD= &QCDm*
        (QpExCD{FLdrate-2} +(100*&drate-FLdrate)*(QpExCD{CLdrate-2} -QpExCD{FLdrate-2}));

* Calculate TotQx -- the total NPV of QALYs lost for this age group;
    TotQCaI=NEV_CaI*QpE_CaI;
    TotQCaD=NEV_CaD*QpE_CaD;
    TotQDev=NEV_Dev*QpE_Dev;
    TotQRep=NEV_Rep*QpE_Rep;
    TotQCIm=NEV_CIm*QpE_CIm;
    TotQCIs=NEV_CIs*QpE_CIs;
    TotQCD =NEV_CD *QpE_CD;

* format calculated output quantities;
```

```
       format
           risk_CaI NEV_CaI QpE_CaI TotQCaI
           risk_CaD NEV_CaD QpE_CaD TotQCaD
           risk_Dev NEV_Dev QpE_Dev TotQDev
           risk_Rep NEV_Rep QpE_Rep TotQRep
           risk_CIm NEV_CIm QpE_CIm TotQCIm
           risk_CIs NEV_CIs QpE_CIs TotQCIs
           risk_CD  NEV_CD  QpE_CD  TotQCD
           e9.;

    * End of data step to calculate benefits by exposure age;
        run;

    * Sum event counts and QALY costs over age groups;
    proc means noprint data=benefits;
        var
           TotQCaI NEV_CaI
           TotQCaD NEV_CaD
           TotQDev NEV_Dev
           TotQRep NEV_Rep
           TotQCIm NEV_CIm
           TotQCIs NEV_CIs
           TotQCD  NEV_CD;
        output out=sumben sum =
           TotQCaI NEV_CaI
           TotQCaD NEV_CaD
           TotQDev NEV_Dev
           TotQRep NEV_Rep
           TotQCIm NEV_CIm
           TotQCIs NEV_CIs
           TotQCD  NEV_CD;
        run;

    * Store:;
    *   TechFlag -- Flag indicating baseline (1) or baseline + supplemental (2) technology;
    *   RunNum   -- Run number;
    *   Values for uncertain parameters as well.;
    data sumben; set sumben;
        TechFlag=&Tech;
        RunNum = &iter;
        cc       = &cc;
        CR1      = &CR1;
        CR2      = &CR2;
        UnHtM    = &UnHtM;
        Cinf     = &Cinf;
        CImP     = &CImP;
        CIsP     = &CIsP;
        CDP      = &CDP;
        SF1Ca    = &SF1Ca;
        SF2Ca    = &SF2Ca;
        SF1Re    = &SF1Re;
        SF2Re    = &SF2Re;
        SF1De    = &SF1De;
        SF2De    = &SF2De;
        CaDP     = &CaDP;
        drate    = &drate;
        QCaIM    = &QCaIM;
        QCaDM    = &QCaDM;
        QDevM    = &QDevM;
        QRepM    = &QRepM;
        QCImM    = &QCImM;
        QCIsM    = &QCIsM;
        QCDM     = &QCDM;
        run;

    * Append record onto dataset results.;
        proc append base=results data=sumben force;

%mend maincalc;


*==============================================================================;
* summary                                                                      ;
```

```
* Summarizes QALY costs and NEV (numbers of events) for the baseline technology  ;
* and for the baseline+supplemental technology.  The results dataset contains    ;
* this information in pairs of records -- each pair corresponding to a single    ;
* draw of random parameter values.                                                ;
* To quantify the incremental change in QALYs and NEVs, the routine collapses    ;
* each pair of records in the results dataset into one record and computes       ;
* differences between each pair of corresponding fields.                         ;
*=================================================================================;
%macro summary;

    * Compute total QALY costs...;
    data Results; set Results;
        TotQ = TotQCaI+TotQCaD+TotQDev+TotQRep+TotQCIm+TotQCIs+TotQCD;
    run;

    * Split the results dataset into two data sets -- one for the baseline technology;
    * results (Tech1) and one for the baseline + supplemental technology results (Tech2);
    data Tech1 Tech2; set Results;
        if TechFlag=1 then output Tech1; else
        if TechFlag=2 then output Tech2;
    run;

    * Merge Tech1 and Tech2;
    proc sort data=Tech1; by RunNum; run;
    proc sort data=Tech2; by RunNum; run;

    data Both (drop = TotQCaI NEV_CaI
                      TotQCaD NEV_CaD
                      TotQDev NEV_Dev
                      TotQRep NEV_Rep
                      TotQCIm NEV_CIm
                      TotQCIs NEV_CIs
                      TotQCD  NEV_CD
                      TotQ);
               merge
               Tech1 (rename=(TotQCaI=TotQCaI1 NEV_CaI=NEV_CaI1
                              TotQCaD=TotQCaD1 NEV_CaD=NEV_CaD1
                              TotQDev=TotQDev1 NEV_Dev=NEV_Dev1
                              TotQRep=TotQRep1 NEV_Rep=NEV_Rep1
                              TotQCIm=TotQCIm1 NEV_CIm=NEV_CIm1
                              TotQCIs=TotQCIs1 NEV_CIs=NEV_CIs1
                              TotQCD =TotQCD1  NEV_CD =NEV_CD1
                              TotQ=TotQ1))
               Tech2 (rename=(TotQCaI=TotQCaI2 NEV_CaI=NEV_CaI2
                              TotQCaD=TotQCaD2 NEV_CaD=NEV_CaD2
                              TotQDev=TotQDev2 NEV_Dev=NEV_Dev2
                              TotQRep=TotQRep2 NEV_Rep=NEV_Rep2
                              TotQCIm=TotQCIm2 NEV_CIm=NEV_CIm2
                              TotQCIs=TotQCIs2 NEV_CIs=NEV_CIs2
                              TotQCD =TotQCD2  NEV_CD =NEV_CD2
                              TotQ=TotQ2)
                       drop = cc CR1 CR2 UnHtM
                              Cinf CImP CIsP CDP
                              SF1Ca SF2Ca SF1Re SF2Re SF1De SF2De CaDP
                              drate
                              QCaIm QCaDM QDevM QRepM QCImM QCIsM QCDM)
                              ;
               by RunNum;


        array Q1{8}     TotQCaI1 TotQCaD1 TotQDev1 TotQRep1 TotQCIm1 TotQCIs1 TotQCD1 TotQ1;
        array NEV1{7}   NEV_CaI1 NEV_CaD1 NEV_Dev1 NEV_Rep1 NEV_CIm1 NEV_CIs1 NEV_CD1;

        array Q2{8}     TotQCaI2 TotQCaD2 TotQDev2 TotQRep2 TotQCIm2 TotQCIs2 TotQCD2 TotQ2;
        array NEV2{7}   NEV_CaI2 NEV_CaD2 NEV_Dev2 NEV_Rep2 NEV_CIm2 NEV_CIs2 NEV_CD2;

        array QD{8}     TotQCaID TotQCaDD TotQDevD TotQRepD TotQCImD TotQCIsD TotQCDD TotQD;
        array NEVD{7}   NEV_CaID NEV_CaDD NEV_DevD NEV_RepD NEV_CImD NEV_CIsD NEV_CDD;

        label TotQCaID = 'QALYs: Cancer Illness';
        label TotQCaDD = 'QALYs: Cancer Death';
        label TotQDevD = 'QALYs: Developmental';
        label TotQRepD = 'QALYs: Reproductive';
```

```
    label TotQCImD = 'QALYs: Crypto - Mild Illness';
    label TotQCIsD = 'QALYs: Crypto - Mod to Sev Illness';
    label TotQCDD  = 'QALYs: Crypto - Death';
    label TotQD    = 'QALYs: All Health Effects';
    label NEV_CaID = 'Num Events: Cancer Illness';
    label NEV_CaDD = 'Num Events: Cancer Death';
    label NEV_DevD = 'Num Events: Developmental';
    label NEV_RepD = 'Num Events: Reproductive';
    label NEV_CImD = 'Num Events: Crypto - Mild Illness';
    label NEV_CIsD = 'Num Events: Crypto - Mod to Sev Illness';
    label NEV_CDD  = 'Num Events: Crypto - Death';

* Take the difference between Tech1 and Tech2 costs and NEVs.  A positive result;
* means Tech1 costs are higher -- i.e., Tech2 has an incremental benefit;
    do i=1 to dim(QD);
        QD{i}=Q1{i}-Q2{i};
    end;
    do i=1 to dim(NEVD);
        NEVD{i}=NEV1{i}-NEV2{i};
    end;

* Compute net present value of QALY costs over the plant lifetime,;
* taking into account the size of the population.;
    do i=1 to dim(QD);
         Q1{i}=(1/(1+drate)**0.5)*&Popsize*(1-(1/(1+drate)**&plantL))*(Q1{i}/drate);
         Q2{i}=(1/(1+drate)**0.5)*&Popsize*(1-(1/(1+drate)**&plantL))*(Q2{i}/drate);
         QD{i}=(1/(1+drate)**0.5)*&Popsize*(1-(1/(1+drate)**&plantL))*(QD{i}/drate);
    end;

* Compute total number of health events -- Taking into account;
* plants lifetime and size of population.;
    do i=1 to dim(NEVD);
         NEV1{i}=&PlantL*&PopSize*NEV1{i};
         NEV2{i}=&PlantL*&PopSize*NEV2{i};
         NEVD{i}=&PlantL*&PopSize*NEVD{i};
    end;

* Compute incremental technology cost;
    IncTech = (&T2_Cap + (1-(1/(1+drate)**&plantL))*(&T2_Op/drate))-
              (&T1_Cap + (1-(1/(1+drate)**&plantL))*(&T1_Op/drate));

* Compute incremental CE of the supplemental technology ($ per NPV QALY);
    CE = IncTech/TotQD;

    format  TotQCaI1 TotQCaD1 TotQDev1 TotQRep1 TotQCIm1 TotQCIs1 TotQCD1 TotQ1
            NEV_CaI1 NEV_CaD1 NEV_Dev1 NEV_Rep1 NEV_CIm1 NEV_CIs1 NEV_CD1

            TotQCaI2 TotQCaD2 TotQDev2 TotQRep2 TotQCIm2 TotQCIs2 TotQCD2 TotQ2
            NEV_CaI2 NEV_CaD2 NEV_Dev2 NEV_Rep2 NEV_CIm2 NEV_CIs2 NEV_CD2

            TotQCaID TotQCaDD TotQDevD TotQRepD TotQCImD TotQCIsD TotQCDD TotQD
            NEV_CaID NEV_CaDD NEV_DevD NEV_RepD NEV_CImD NEV_CIsD NEV_CDD

            IncTech CE

            e9.;

* Finish creation of database containing summary of both baseline and baseline +;
* supplemental technology (dataset Both);
    run;

*************************************************************;
* Summarize results (use either proc means or proc univariate);
*************************************************************;
proc means data=Both mean stderr;
    var TotQCaI1 TotQCaD1 TotQDev1 TotQRep1 TotQCIm1 TotQCIs1 TotQCD1 TotQ1
        NEV_CaI1 NEV_CaD1 NEV_Dev1 NEV_Rep1 NEV_CIm1 NEV_CIs1 NEV_CD1;
run;
proc univariate data=Both noprint;
    var TotQCaI1 TotQCaD1 TotQDev1 TotQRep1 TotQCIm1 TotQCIs1 TotQCD1 TotQ1
        NEV_CaI1 NEV_CaD1 NEV_Dev1 NEV_Rep1 NEV_CIm1 NEV_CIs1 NEV_CD1;
    output out=stats
        median = TQ1md  TQ2md  TQ3md  TQ4md  TQ5md  TQ6md  TQ7md TQ8md
```

```
                    NEV1md NEV2md NEV3md NEV4md NEV5md NEV6md NEV7md
            p90    = TQ1p90  TQ2p90  TQ3p90  TQ4p90  TQ5p90  TQ6p90  TQ7p90 TQ8p90
                    NEV1p90 NEV2p90 NEV3p90 NEV4p90 NEV5p90 NEV6p90 NEV7p90;
    run;
    proc transpose data=stats out = tr_stats;
    proc print data=tr_stats label; run;

    proc means data=Both mean stderr;
        var TotQCaI2 TotQCaD2 TotQDev2 TotQRep2 TotQCIm2 TotQCIs2 TotQCD2 TotQ2
            NEV_CaI2 NEV_CaD2 NEV_Dev2 NEV_Rep2 NEV_CIm2 NEV_CIs2 NEV_CD2;
    run;
    proc univariate data=Both noprint;
        var TotQCaI2 TotQCaD2 TotQDev2 TotQRep2 TotQCIm2 TotQCIs2 TotQCD2 TotQ2
            NEV_CaI2 NEV_CaD2 NEV_Dev2 NEV_Rep2 NEV_CIm2 NEV_CIs2 NEV_CD2;
        output out=stats
            median = TQ1md  TQ2md  TQ3md  TQ4md  TQ5md  TQ6md  TQ7md TQ8md
                    NEV1md NEV2md NEV3md NEV4md NEV5md NEV6md NEV7md
            p90    = TQ1p90  TQ2p90  TQ3p90  TQ4p90  TQ5p90  TQ6p90  TQ7p90 TQ8p90
                    NEV1p90 NEV2p90 NEV3p90 NEV4p90 NEV5p90 NEV6p90 NEV7p90;
    run;
    proc transpose data=stats out = tr_stats;
    proc print data=tr_stats label; run;

    proc means data=Both mean stderr;
        var TotQCaID TotQCaDD TotQDevD TotQRepD TotQCImD TotQCIsD TotQCDD TotQD
            NEV_CaID NEV_CaDD NEV_DevD NEV_RepD NEV_CImD NEV_CIsD NEV_CDD;
    run;
    proc univariate data=Both noprint;
        var TotQCaID TotQCaDD TotQDevD TotQRepD TotQCImD TotQCIsD TotQCDD TotQD
            NEV_CaID NEV_CaDD NEV_DevD NEV_RepD NEV_CImD NEV_CIsD NEV_CDD;
        output out=stats
            median = TQ1md  TQ2md  TQ3md  TQ4md  TQ5md  TQ6md  TQ7md TQ8md
                    NEV1md NEV2md NEV3md NEV4md NEV5md NEV6md NEV7md
            p90    = TQ1p90  TQ2p90  TQ3p90  TQ4p90  TQ5p90  TQ6p90  TQ7p90 TQ8p90
                    NEV1p90 NEV2p90 NEV3p90 NEV4p90 NEV5p90 NEV6p90 NEV7p90;
    run;
    proc transpose data=stats out = tr_stats;
    proc print data=tr_stats label; run;

    proc univariate data=Both;
        var CE;
    run;

    *************************;
    * Sensitivity analysis...;
    *************************;
    * Omitted variables that are now constant: CadP CR1 drate;
    proc glm data=both;
         model CE = cc CR2 UnHtM
                    Cinf CImP CIsP CDp
                    SF1Ca SF2Ca SF1Re SF2Re SF1De SF2De
                    QCaIm QCaDM QDevM QRepM QCImM QCIsM QCDM;
    run;

%mend summary;

* Set up dummy database with one record that can be read in to establish the;
* results data set.;
    data dummy;
         input foo $;
         cards;
         0
         ;
    run;

%Simulate; run;
```