**South Fork Research, Inc**

**ISEMP Watershed Model**

**Habitat Modification Effects on Salmonid Population Dynamics**

**Matt Nahorniak**

Contents

**[1.](#_Toc479584367)****[Introduction](#_Toc479584367)** [3](#_Toc479584367)

**[2.](#_Toc479584368)****[System Requirements](#_Toc479584368)** [4](#_Toc479584368)

**[3.](#_Toc479584369)****[Files Included in the Watershed Model](#_Toc479584369)** [4](#_Toc479584369)

**[4.](#_Toc479584370)****[The Beverton Holt Spawner Recruit Model](#_Toc479584370)** [6](#_Toc479584370)

**[5.](#_Toc479584371)****[Tracking Spawners and Recruits for Run Reconstruction](#_Toc479584371)** [18](#_Toc479584371)

**[6.](#_Toc479584372)****[Stochasticity in the Watershed Model](#_Toc479584372)** [20](#_Toc479584372)

**[7.](#_Toc479584373)****[Including Time Based Changes for Input Parameters](#_Toc479584373)** [27](#_Toc479584373)

**[8.](#_Toc479584374)****[Entering Simulation Data and Parameter Estimates into the Input Files](#_Toc479584374)** [32](#_Toc479584374)

**[9.](#_Toc479584375)****[Running the Model in R](#_Toc479584375)** [53](#_Toc479584375)

**[10.](#_Toc479584376)****[Model Output](#_Toc479584376)** [54](#_Toc479584376)

**[11.](#_Toc479584377)****[Checking the Inputs with Graphs](#_Toc479584377)** [59](#_Toc479584377)

**[12.](#_Toc479584378)****[Common Problems and Troubleshooting](#_Toc479584378)** [61](#_Toc479584378)

**[13.](#_Toc479584379)****[References](#_Toc479584379)** [63](#_Toc479584379)

1. **Introduction**

A model for salmon population dynamics, as described in “Salmon ISEMP Watershed Model Development: Adding Stochasticity to the Life History Model Structure” (QCI 2006), and as describe by Sharma et al (Yuen and Sharma, 2005), has been coded in the R programming language. This model, an improvement and enhancement of the Visual Basic QCI 2006 model, provides an analytical framework for habitat action effectiveness monitoring.

The model implements the Beverton-Holt spawner-recruit salmon population dynamics model (Beverton and Holt 1957). Inputs describing one or more sites within a watershed, survival estimates by life stage, etc., are user specified model inputs, as are measures of uncertainty in parameter estimates, and estimates of natural parameter spatial, temporal, and pure variability. Initial salmonid populations, by life stage, are also user specified. The model calculates fish populations by life stage over for each subsequent year up to a user specified number of years. Hatchery fish introductions into a watershed, and parameters describing the relative robustness and fecundity of hatchery fish and their descendants, can also be user specified.

Included in the model is the option for user specified levels of stochasticity, applied at various levels. This stochasticity serves two functions: 1) estimation of uncertainty of model results stemming from *uncertainty* of input parameters; and 2) estimation of temporal, spatial, and pure variability in the results stemming from temporal, spatial, and pure variability in the input parameters. Stochasticity at all levels is user specified, and structured so as to give rise to natural correlations among input parameters. These correlation structures enable a stochastic model much more reflective of natural processes than could be achieved by assuming independence across all parameters.

Also included in the model is the ability to include time based trends and/or step function changes for all user specified parameters. Such changes may reflect, for example, changes in watershed management that lead to gradual increases in forested lands within a watershed, or discrete changes, such as a change in dam management, leading to a step function shift in seasonal water flows.

Multiple sites may be modeled simultaneously. “Sites”, in this model, refer to a user-defined spatial scale over which the user wishes to define the input parameters. A site may be a reach within a tributary, a tributary within a watershed, a watershed within a subbasin, etc. The advantages of concurrent modeling of multiple sites, as opposed to modeling one site at a time, are two-fold: First, sites within a watershed are likely not independent; a low water year for a single site is likely a low water year for all sites within a watershed; this model can account for such correlations. Secondly, modeling multiple sites concurrently allows for inclusion of cross-site migration, where fish at various life stages have some user specified non-zero probability of migrating to a different site within a watershed. In addition, modeling multiple sites concurrently allows summarization of results at whatever spatial level he or she chooses, after the completion of the simulation (i.e. results may be summarized by site, stream, river, watershed, etc).

The model has been structured such that it is flexible enough to handle different species of salmonids (though they cannot be modeled simultaneously). This includes steelhead, which are generally more complex than other salmonid species, as far as modeling the transitions from one life stage to the next.

1. **System Requirements**

The Watershed model has been written in the R programming language. R is a free, open source programming language designed specifically for statistical computing and graphics. The software and documentation are available free at: <http://www.r-project.org/>.

Numerous add-on packages are available within the R environment. This program requires the packages “MCMCpack”, and “VGAM”. The “MCMCpack” package provides functions relevant for the Dirichlet distribution, and the “VGAM” package provides for positive only normal distribution functions (a modification to the Normal distribution that enforces a positive only values rule).

To install the required add-on packages, start R, click on “install package” and select a CRAN mirror site (any site should work). Scroll down and select the package “MCMCpack”, after which installation will take place automatically. Installation of this package need only be done once for the computer on which R is loaded. Repeat this process for the “VGAM” package.

1. **Files Included in the Watershed Model**

In addition to this user’s guide document, included in the Watershed model are the following files. Details on each are included in sections to follow.

**R-Files**

* **Watershed.R:** The main R script, run by the user, that calls all other functions used to perform the watershed simulation.
* **Watershed\_ReadData.R**: R file containing functions used to read data from the various Excel input files
* **Watershed\_MonteCarlo.R**: R-code containing functions used to add stochasticity to the watershed model
* **Watershed\_BevHolt.R**: R-code containing functions to implement the Beverton Holt Algorithm
* **Watershed\_PostProcessing.R**: R-code containing function to create plots and write output files.
* **Watershed\_GraphInputs**.**R**: R-code that is NOT part of the Watershed Model per se. This code can be used independently, along with the input files, to create plots of all the input variables versus simulation time. This can be used to check inputs.

**Input Data File Templates**

Input parameters, initial conditions, and simulations details are input to the model via a set of comma separated value (.csv) files. .CSV files are easily and quickly read (and written ) by R-code. A Downside of using .csv files, as opposed to Microsoft Excel files, is that cell formatting (column widths, cell colors, etc.) are not available for .csv files. To aid in the creation of input files, a set of formatted Microsoft Excel files is also available, in which the user may enter input data and then save the completed input files as .csv files. Alternately, the user may work with .csv files directly.

* **Header File** (provided as “Header\_File.csv”, or the Excel template as “Header\_File\_Template.xlsx”.) This file provides a template for the user to enter high level data inputs for the simulation, as described in section 7.1.
* **Site Level Input File** (provided as “Site1\_T1.csv” or the Excel template “Input\_Template.xlsx”) This file provides a template for detailed user inputs for required modeling parameters such as fish survival probabilities by life stage, land use and habitat information, and inputs for temporal trends; as well as input describing stochasticity of these parameters. Details on these inputs are found in sections 7.2 and 7.3. Multiple copies, saved under separate filenames, of this file may be used to when modeling step changes in input parameters. At least one site level input file is required for each site included in the watershed model.
* **Initial Values File** **(**provided as “Site1\_Inits1.csv”, or the Excel template "Initial Values Template.xlsx”. ) These files are used to specify Initial values for fish counts, by life stage. One file is required for each site modeled. Details on providing inputs to this file are found in section 7.4.
* **Cross Site Migration File** (provided as Cross\_Site\_Migration\_T1.csv” or the Excel template “Cross\_Site\_Migration\_Template.xlsx”. ) This file allows the user to specify the probability of a fish migrating from one site to another within a watershed, by life stage. For example, while a spawner generally returns to the same site from which it smolted, the user may specify a probability that it returns to a different site to spawn.

1. **The Beverton Holt Spawner Recruit Model**

The watershed population model follows the Beverton-Holt spawner recruit model (Beverton and Holt 1957) as has previously been implemented for modeling life stage population dynamics for salmonid populations (Yuen and Sharma, 2005; Mousalli and Hilborn, 1986). The basic structure of the Beverton Holt model is as follows:

(4.1)

where Ni,t is the number of individuals alive at the beginning of life history stage i at time t for site k. pk,i,t is the “productivity” at stage i (the maximum survival rate from stage i to i +1) and ck,i,t is the “capacity” (the maximum number of individuals that could survive from stage i at time t to stage i+1 at time t+1). Note that within the salmon model, certain life stages occur within the same year as previous life stages, while others occur the following year. Thus, the subscript on t in the above equation will sometimes be the same on the left and right sides of the equation, while in other cases it will be incremented by one. (See details below).

Productivity at time t for site k is modeled as:

 (4.2)

where:

pk,I,t = density independent productivity for stage i dependent on the relative importance / relationship between productivity and land use in that stream (QCI 2006).

Ei,q= Scalar showing the importance of land-use type (q) for overall productivity (see Table 2-2 in QCI 2006). Note: In the R code, the productivity variable is called “Prod\_Scalar [k,i,t]” instead of “E[k,i,t]”.

Srk,i,t = site and time specific average maximum survival rate from one stage to the next in the fresh-water life history of the species given average conditions (see Table 2-5 of QCI 2006) under a baseline in the best possible habitat suited for their survival.

Capacity at time t for site k is at life stage i is modeled as:

(4.3)

Where Hk,j,t is the amount of habitat type j in watershed k at time t, and Dk,j,i,t is the maximum density, in fish per unit area area, in square meters, of habitat type j at site k, during life stage i, at time t. Hk,j,t is modeled as:

(4.4)

which can be re-written as

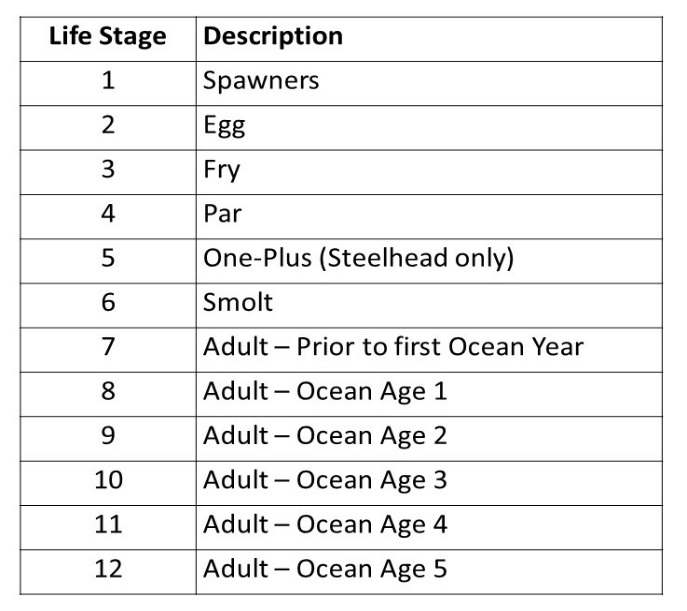
(4.5)

The product is entered as a single user input (as shown in table 2-2 of QCI, 2006).

is the proportion of each habitat type j in each land use category, q, at time t for site k.

While this document is not intended as a comprehensive review of the Beverton Holt Spawner Recruit Model, sections 4.1 through 4.10 detail specifically how this watershed model, as programmed in R, implements the Beverton Holt algorithm at each life stage. In all cases, the index “t” refers to a year. Certain life stages are dependent on a prior life stage within the same year. Other life stages are dependent on the prior life stage during the previous year. Nomenclature for this section is in the same format as is used in the R-code in which this algorithm is implemented. Subscripting is not used within R code. Instead, indexes appear within brackets. For example, N\_FISH[k, i, t] refers to the salmon population at site k for life stage i and time t. The life stages for i=1 to 12 are shown table 4.1:

Table 4.1. Salmonid Life Stages Modeled



Only the implementation of the Beverton Holt model for One-Plus (section 4.5), and the implementation of a cross site migration model (section 4.11) are unique to this implementation of the Beverton Holt population dynamics model. The portions of the model unique to this implementation deal with the added complexities of steelhead modeling, especially during the One-Plus life stage (life stage 5).

* 1. **Number of Spawners at time t**

Modeling fish populations starts with an assumed number of fish, by life stage, by site, at time zero. The initial populations are user specified (see section 7.3). The number of spawners at time t+1, as well as the average fecundity of these spawners, will be the last in the series of Beverton Holt calculations performed for each time step. See section 4.10 for these calculations.

* 1. **Life Stage 2: Eggs at time t**

The number of eggs at year t is a function of the number of spawners at year t, the spawner capacity of the site, and the productivity. Unlike other life stages, the productivity (fecundity, in this case) will be dependent on the distribution of ages of the mature fish that return to spawn. Fecundity by age is user specified. Details on this are described in section 4.10. The Beverton Holt equation, used to calculate the number of eggs at time t, for site k, is implemented as follows:

N\_FISH[k,2,t]= N\_FISH[k,1,t] / (1/p[k,1,t]+ 1/c[k,1,t] \* N\_FISH[k,1,t]) (4.6)

Note that the “2” refers to the 2nd life stage (egg) as specified in table 4.1.

* 1. **Life Stage 3: Fry at time t+1**

Fry are produced from the previous year’s eggs. The number of fry at time t+1 for site k is calculated, via the Beverton Holt model, as:

N\_FISH[k,3,t+1] = N\_FISH[k,2,t] / (1/p[k,2,t]+ 1/c[k,2,t] \*N\_FISH[k,2,t]) (4.7)

* 1. **Life Stage 4: Parr at time t**

The number of parr are calculated from the current year’s fry as:

N\_FISH[k,4,t] = N\_FISH[k,3,t] / (1/p[k,3,t]+ 1/c[k,3,t] \*N\_FISH[k,3,t]) (4.8)

* 1. **Life Stage 5: One-Plus at time t**

Steelhead, unlike other species, may remain in freshwater for years before they smolt (attempt to migrate to the ocean). Or they may never migrate to the ocean, and thus become “resident rainbows”. Enabling tracking for the generalized case requires some additional modeling steps, as well as additional user inputs. We deal with this complexity by allowing multiple years of freshwater residency as “One-Plus.”

One-Plus at time t may come from two “sources”: parr that have survived into One-Plus, or One-Plus from previous years that have neither migrated to the ocean nor returned to spawn as “resident rainbows.” Species other than Steelhead generally smolt (attempt their migration to the ocean) with 100% probability at the next time step after becoming a One-Plus. Steelhead and other species are modeled via the same code in the Watershed model; the differences are accounted for in the input file by specifying a Species in the first row of the input file (see section 6). By specifying a species of “Chinook”, the One-Plus algorithm is implicitly skipped.

The remainder of this description of the One-Plus life stage assumes the more complex, generalized model, suitable for steelhead modeling.

Keeping track of the number of One-Plus, the age distribution of One-Plus, etc. presents some difficulties. We must assume, for example, that the capacity of a site for One-Plus is reduced if a greater percentage of One-Plus are older and therefore larger. And we must make some assumption about which fish, attempting to enter or remain at the One-Plus life stage, will compete successfully for the limited capacity. The watershed model deals with these complexities in the following manner.

First, a variable N5\_FISH has been created just to account for One-Plus by One-Plus age (i.e. the number of time steps a fish has remained as a One-Plus). N5\_FISH[k, i5, t,] is the number of One-Plus at site k, at “One-Plus age” i5, for year t. (The “5” was used as One-Plus is the 5th life cycle tracked in the model). i5=1 indicates a “new” One-Plus (just progressed to One-Plus from parr), while i5=2 or greater indicate One-Plus that have remained “i5-1” years as One-Plus.

N5.Psmolt[i5] is a user specified input vector indicating the probability that a fish will move from One-Plus to smolt (i.e. attempt to migrate to the ocean) at One-Plus age i5. For non-steelhead, this can simply be set to 1 for i5=1. For steelhead, this may range from 0 to 1.

N5.cap[i5] is a user specified vector of scalars representing the additional capacity per fish required for One-Plus, by One-Plus age, relative to “new” One-Plus. For capacity considerations, a single One-Plus at One-Plus age i5 will be counted as N5.cap[i5] One-Plus. In other words, the actual capacity, as measured in actual number of fish, will be less for a distribution of fish ages than it would be for a population of new One-Plus only. The actual capacity (maximum number of One-Plus a site could hold) will be adjusted based on the additional capacity required for each fish as it grows while staying in the One-Plus life stage. These capacity adjustments are user specified (see section 7).

Just as the capacity requirement may change for One-Plus that remain as One-Plus for multiple years prior to migrating (or dying, or spawning), the probability of surviving from one year to the next is likely to be a function of One-Plus age. SR5[i5] is a vector of survival probabilities for One-Plus, by One-Plus age. Again, this vector is user specified.

With the additional parameters as specified above, Beverton-Holt calculations for the number of One-Plus at time t+1 are implemented as described in sections 4.5.1 through 4.5.5.

**4.5.1. Step 1: Candidate Fish**

The number “candidate” fish at time seeking to become One-Plus at time t+1 includes all fish at life stage 4 (parr) and all One-Plus that do not migrate to the ocean. For capacity calculations, the One-Plus that stay are counted as more than one fish by scaling by age via the values in the vector N5.cap[i5]

CandidateFish = N\_FISH[k,4,t] + sum(N5\_FISH[k,1:10,t]\*(1-N5.Psmolt)\*N5.cap) (4.9)

Scaling by N5.cap results in CandidateFish being counted in terms of “equivalent first year” One-Plus. This concept may be helpful in understanding the logic behind the calculations.

**4.5.2. Step 2: Weighted Average for Candidate Survival Probability**

The probability of survival for these potential One-Plus for time t+1 is taken as a weighted average of the survival probabilities, of the parr and One-Plus at time t, seeking to become (or remain) One-Plus at time t+1:

SR5.Candidate = (N\_FISH[k,4,t]\*Sr[4]+sum(N5\_FISH[k,,t]\*(1-N5.Psmolt)\*SR5))/

(N\_FISH[k,4,t]+sum(N5\_FISH[k,,t]\*(1-N5.Psmolt))) (4.10)

It is seen in, Eq. 4.10, that the survival probabilities are weighted by the number of candidate fish vying to become or stay as One-Plus.

**4.5.3. Step 3: Weighted Average for Candidate Productivity**

The productivity is then calculated in the normal manner, using the weighted survivability from above:

pN5 = SR5.Candidate\*(sum(Prod\_Scalar[,5]\*L[k,,t])/sum(L[k,,t])) (4.11)

Note that the productivity scalar (“E” in QCI 2006) is assumed to not be a function of the age distribution of One-Plus.

**4.5.4. Step 4. Equivalent first year One-Plus**

The total number of “equivalent first year” One-Plus is then calculated based on the Beverton Holt model. These are “equivalent first year” rather than actual, because older fish will count as more than one first year equivalent fish.

N\_FISH[k,5,t+1] = CandidateFish / (1/pN5 + 1/c[k,5,t] \*CandidateFish) (4.12)

**4.5.5. Calculate Number of One-Plus by Age**

At this point, the difficulty lies in figuring out, of the equivalent first year fish, how many are at each One-Plus age (i5) and what the actual numbers of each are. To do this, we will assume fish survive into the next year’s One-Plus population in proportion to the product of the number of fish attempting to enter each One-Plus age (i5) and the survival probability for candidate group trying to reach One-Plus age. N5temp is a temporary variable used to calculate relative frequencies of One-Plus by One-Plus age, which is then scaled such that the total first year equivalent fish in N5temp is equal to N\_FISH[k, 5, t+1] while retaining the relative distribution of ages.

Since for i5=1, the candidate fish are maturing parr, N5temp[1] is as follows:

N5temp[1] = N\_FISH[k,4,t]\*Sr[4] (4.13)

For the remaining One-Plus ages, the candidate fish come from One-Plus from the prior year:

N5temp[i5] = N5\_FISH[k,i5-1,t]\*(1-N5.Psmolt[i5-1])\*SR5[i5-1] (4.14)

N5 temp contains the appropriate ratios for each One-Plus age, but must be scaled such that the total equivalent first year One-Plus match that calculated by Eq. 4.12. These values are then stored as N5 at time t+1 by:

N5\_FISH[k,,t+1]= N5temp \*(N\_FISH[k,5,t+1]/((sum(N5temp))\*N5.cap)) (4.15)

Note: For non-steelhead species, all of the above equations and calculations are carried out; however, the probability of transitioning from One-Plus to smolt at One-Plus age 1 will be 1, and the result of the calculations are equivalent to those that would be produced from a simple Beverton-holt model for life stage 5; as follows:

N\_FISH[k,5,t+1] = N\_FISH[k,4,t] / (1/p[k,4,t]+ 1/c[k,4,t] \*N\_FISH[k,4,t]) (4.16)

Thus we have the completed estimate for One-Plus population at time t+1.

In Steelhead modeling, there is also some non-zero probability that a One-Plus will never migrate to the ocean, and instead spawn after some number of years as a One-Plus. These are termed “resident rainbows”. In order to model this, there is a user specified probability that a One-Plus of a given One-Plus age will spawn (instead of either remaining as a One-Plus or migrating to the ocean). At the conclusion of the One-Plus calculations, a fraction of One-Plus, at each One-Plus age i5, are “removed” from the population and designated spawners, according the following:

RainbowSpawners[i5] = N5\_FISH[k,i5,t]\*N5.Pspawn[k,i5,t] (4.17)

This same number of fish is then subtracted from the One-Plus population for each life stage. These resident rainbow spawners will be added to the total number of spawners, and the fecundity of these spawners factored into the total average fecundity of spawners (see section 4.10).

* 1. **Life Stage 6: Smolts at Time t**

Since smolts (fish attempting to migrate to the ocean) arise from a distribution of One-Plus ages (at least for the generalized steelhead model), the number of “candidate” smolts is the sum of the product of the number of One-Plus at each life stage and the probability of smolting, by One-Plus age. This is modeled as follows:

CandidateN6 = sum(N5\_FISH[k,,t]\*N5.Psmolt) (4.18)

From which the normal Beverton-Holt relationship is used to estimate the number of smolts at time t. Note that, at this point, there is no assumed dependence for capacity or production on the distribution of the ages of One-Plus advancing to smolts.

N\_FISH[k,6,t] = CandidateN6 / (1/p[k,5,t]+ 1/c[k,5,t] \* CandidateN6) (4.19)

* 1. **Life Stage 7: Ocean Age 0**

Note: Beyond life stage 6, we assume that distribution of One-Plus ages for a given fish population is irrelevant; and the number of years spent as One-Plus is ignored.

The number of smolt successfully migrating to the ocean is essentially based only on the probability of surviving the dam(s) along the way (See QCI 2006, Attachment I, equation 4).

N\_FISH[k,7,t] = N\_FISH[k,6,t] / (1/ + 1/D.dam[k,t] ) (4.20)

It is assumed that that there are no capacity effects for fish that have reached the ocean (QCI, 2006). This assumption is implemented in the model by effectively setting the capacity to infinite such that 1/c is zero.

* 1. **Life Stage 8: Ocean Age 1**

The number of fish surviving to life stage 8 (ocean age 1), is calculated as follows:

N\_FISH[k,8,t+1] = N\_FISH[k,7,t] / (1/o[1] + 1/c[k,7,t] \*N\_FISH[k,7,t]) (4.21)

The variable “o” is used to track the ocean survival probability by ocean age. This is also a user specified vector (see section 7). The model accounts for ocean ages up to 10 years, starting as life stages 8. The life stage index, i, indexes from 1 to 17. Thus the life stage index “i-7” is used as the index for o, and o[k,i-7,t] indexes from o[k,1,t] to o[k,10,t].

* 1. **Life Stages 9-12: Adult Salmon**

Salmon will continue to move from adult ocean age salmon to the next age of salmon according to survival (o) for the given life stage i, except that a certain fraction at each ocean age will mature and, rather than remain in the ocean to reach the next ocean age, will return to the watershed to attempt to spawn. Mat8Plus is a vector of probabilities that a fish of ocean age i-7 will “mature” (return to spawn) at life stage i (i.e. ocean age i-7). Mat8Plus is a vector user specified values for each site.

N\_FISH[k,i+1,t+1] = (N\_FISH[k,i,t]\*(1-Mat8Plus[k,i-7,t])) /

(1/o[k,i-7,t] + 1/c[k,i,t] \*N\_FISH[k,i,t]\*(1-Mat8Plus[i-7])) (4.22)

* 1. **Spawners: Adult to Life Stage 1:**

As described above, spawners come from the portion of adult fish that reach maturation at a given ocean age i-7. The variable “NT” is created to track potential spawners by ocean age (i-7).

NT[k,i-7,t+1] = (N\_FISH[k,i,t]\*(Mat8Plus[k,i-7,t])) /

(1/o[i-7] + 1/c[k,i,t] \*N\_FISH[k,i,t]\*(Mat8Plus[k,i-7,t])) (4.23)

The total number of spawners is thus the sum of NT, less the fraction harvested; as well as the number of spawning “resident rainbows” (see section 4.5):

N\_FISH[k, 1, t+1] = sum(NT[k,,t+1]) \* (1-harvest.wild[k,t])

+ sum(RainbowSpawners) (4.24)

Note that for non-steelhead modeling, the probability of a “resident rainbow” (a One-Plus that spawns without having migrated to the ocean) will be user specified as zero.

Finally, the fecundity of the spawners is a weighted average of the fecundity of the spawning fish (by ocean age and resident rainbow age), as follows:

Fecund[k,t+1] = (sum(NT[k,,t+1]\*Fc.by.O.Age[k,,t+1]) + sum(RainbowSpawners\*N5.Rainbow.Fecundity[k,,t+1])) / (sum(NT[k,,t+1])+sum(RainbowSpawners)) (4.25)

The fecundity by ocean age (for returning Steelhead), and for “One-Plus age” (for resident rainbows) are user specified values in the input files (see section 7.2).

The fecundity at time t+1 will factor into the site productivity for spawner to eggs for the next time cycle, as the survival parameter on a per-fish basis.

Note that since Fecundity is on a **per-fish**, rather than per-female fish, basis, the numbers in the input file need to **be specified as per-fish, not per-female** (See section 7).

* 1. **Cross-Site Migration Model**

When more than one site is modeled in parallel, the Watershed model assumes there is a non-zero probability that a fish at a given life stage will emigrate from one site to another site. A user input file is used to specify these probabilities, and after each iteration of the full Beverton-Holt sequence (across all life stage), a fraction of the populations at certain life stages is re-assigned to another site based on these probabilities. Matrices describing the fraction of fish that migrate from each site to each other site, by life stage, are user specified (see section 7).

* 1. **Hatchery Effect**

Because hatchery management can have a significant effect on both hatchery and wild fish populations, and wild fish interact with hatchery fish in multiple ways over a life cycle, the option to model hatchery introduced fish concurrently with wild fish is included in this model.

Many Beverton-Holt life cycle models integrate hatchery fish into the model by tracking the overall degradation to a population’s ability to compete and survive at various life stages, as hatchery fish are introduced and interbreed with the native population [ref 6]. This model makes similar assumptions regarding the effects of hatchery introduced fish, but uses a different approach to integrate hatchery effects into the life cycle model. Rather than quantifying the degradation of an entire population as hatchery fish interact with wild fish, various genetic subpopulations are tracked independently, and user specified relative survival and fecundity parameters are used to model relative survival as the different subpopulations compete for limited capacity.

Specifically, the subpopulations modeled include: wild fish, hatchery bred fish, and all offspring combinations of wild and natural fish up to three generations removed from hatchery origin. Any fish at least three generations removed from hatchery origin is, for modeling purposes, considered wild. Unique survival probability scalars by life stage, and unique fecundity scalars, are user defined for each of these possible genetic subpopulations. Table 4.12.1 illustrates the 11 possible genetic sub-populations. Fish labeled “N” are considered natural, or “wild” fish; while H1 are hatchery bred fish, where “H” refers to hatchery, and “1” indicates “first generation”. If an H1 breeds with another H1 in the wild, the offspring is referred to as “H2”, meaning pure hatchery origin and 2nd generation from the hatchery. If an H1 breeds with an N, the offspring are referred to as “N-H2” (note that there is no need to indicate a generation number tied to the “N”). The following table lists all possible genetic combinations up the three generations from the hatchery. Beyond any third generation hatchery element, the fish is assumed wild; thus an H3 becomes an “N” in the following generation, and all fish can be categorized into one of the eleven categories below.

**Table 4.12.1 Wild, Hatchery, and all other combinations of genetic subspecies Tracked in Model**



Also included in the preceding table is a percent genetic makeup of natural, H1, H2, and H3 brood stock, as well as an indication of “average hatchery generations” (AHG) where 1 indicates hatchery born and 0 indicates purely wild (or > 3 generations removed from any hatchery ancestry). These values are not used directly in the model, but may be of interest to the user when specifying relative survival and fecundity parameters for various genetic subspecies.

The advantage of specifying genetic subspecies as in table 4.12.1 enables integration of hatchery effects into the current model, including simulations where all the additional complexities of steelhead (variable years as One-Plus) are included, as well as cross site migration probabilities, variable years spent as adult prior to spawning, etc., that most Beverton-Holt models simplify or ignore. It also provides maximum flexibility to the user to specify relative survival probabilities by life stage / genetic subspecies category, as well as fecundity by genetic subspecies category; and to update and modify these specifications as understanding improves. The disadvantages of modeling hatchery fish in this manner include putting more impetus on the user for input rather than relying on a specific model for genetic degradation, as well as limiting the genetic variation to only three generations removed from hatchery fish.

Within the model, the various genetic subspecies populations interact and compete for the given capacity at each life stage. For modeling purposes, this is done on a per-life stage basis by first calculating a weighted average survival probability across all genetic groups (weighted by population density of each group), calculating the total number of surviving fish, and then allocating those surviving fish according to the relative survival probabilities of each group. (Note this is nearly identical to how survival of steelhead One-Plus of various ages is modeled, as described in section 4.5, except that there is no assumed capacity usage difference among the different genetic categories).

Note also that in some other models [ref 6], the relative survivability and the relative ability to compete are specified as separate inputs. When assigned on a per-subpopulation level, competing and surviving at a given life stage are essentially the same act, thus only a single value is specified by the user and used in modeling. All user inputs in this case are merely relative survival probabilities, where values are relative to the survival probability of a wild fish at each life stage.

Hatchery effects are also considered when estimating fecundity. For this model, the user will specify the fecundity of female spawners by age of the returning spawners and by the genetic subspecies type. Again, the user specifies fecundity effects relative to wild fish. In addition, the fraction of returning spawners that are female is known to vary across age of returning fish as well as the genetic subspecies type. These values are also user specified in the model.

The model also makes the simplifying assumption that, for those fish that survive to spawn, mate selection occurs at random, with no preferential force driving wild-wild pairings, hatchery-hatchery pairings, etc.

Details on how user specifications of hatchery effects are input into the model are provided in section 7.2.

Apart from introduction, survival, fecundity, and male/female ratio differences between wild and hatchery origin fish, management differences may specify differences in relative harvest rates. This can also be user specified (see section 7.2, figure 7.2.9). In this case, only differences between hatchery origin (first generation, H1 fish) and all other naturally spawning (whether wild or more than a generation removed from the hatchery) are specified, as only hatchery fish (H1) can be uniquely identified at harvest and thus managed separately. This assumes no differences in the likelihood of being harvested, due to differences in fish behavior, across the 10 non-hatchery (H1) subspecies classifications.

1. **Tracking Spawners and Recruits for Run Reconstruction**

The ISEMP Watershed Model tracks spawners and recruits for each year, to be output and/or plotted by an external function. This code is described separately because it has no effect on the working of the model. Although steelhead can spend anywhere from 0 to 6 years as a One-Plus prior to smolting, once the fish have advanced to Life Stage 8 (smolt), there is no assumed dependency on “true age” in the remainder of the model. In order to recreate the data for run reconstruction, we nonetheless have to keep track of true age.

* 1. **Life Stage 6 (Smolt)**

For Steelhead, the array N\_SMOLTS tracks fresh water ages for each time t, site k, and genetic type g.

N\_SMOLTS[k,i5,t,g] = SmoltSurvivors[k,i5,t,g]

For Chinook, all smolt have a fresh water age of 1.

N\_SMOLTS[k,1,t,g] = sum(N\_FISH[k,6,t,g,])

* 1. **Life Stage 7 (New Adults that reached the ocean)**

Once a fish has reached the ocean, the N\_ADULTS array is used to keep track of all ocean fish by fresh water age, and ocean age (0, which is index 1). True age is the sum of these two ages.

N\_ADULTS[k,i5,t,g,1] = N\_SMOLTS[k,i5,t,g] / (1/Sr[k,7,t] ) \* Rel\_Surv[k,6,t,g]

It is assumed that the survival calculation is not dependent on fresh water age (i5).

* 1. **Life Stage 8-12 (Adults in ocean)**

For these life stages, the model has to deal with both escapement (fish leaving the ocean to attempt to spawn) and survival (fish that die). To continue to keep track of adults by age distribution, the following principles apply:

* When calculating the fish that wish to remain in the ocean – staying as adults for at least one more year – the N\_ADULTS array shall maintain the current distribution of fresh water ages.
* When calculating spawners that return to the ocean to live (and potentially survive to spawn again), assume that this cohort preserves the current distribution of fresh water ages.
* When calculating N\_RECRUITS – adults that leave the ocean to try to spawn – maintain the current the distribution of freshwater ages.

1. **Stochasticity in the Watershed Model**

Stochasticity is included the model in the model for two purposes: 1) To quantify *uncertainty* in the results stemming from *uncertainty* in the inputs, and 2) to quantify variability in the results due to *variability* in the input parameters. This variability includes temporal variability, site to site spatial variability, and within site “pure” variability.

Uncertainty in inputs will be referred to as “run to run” variability or uncertainty. Each “run” refers to a single run of the watershed model. The Monte-Carlo simulation will run the model multiple times, each time randomly selecting values of each parameter based on a user specified mean and uncertainty and a fixed distribution type (either normal or Dirichlet, depending on the parameter being specified). Time based variability is referred to as “Year to year” or “temporal” variability. Spatial variability (within each year) is referred to as “site to site” variability. Within site, within year variability is referred to as “within site” or “pure” variability.

Specifying variability components in this way has the advantage of forcing correlation structures in the model, which may have important impacts with respect to the results. These correlation structures undoubtedly better describe reality than a model which fails to account for correlation among variables within a site, within a year, within a run, etc. Models that fail to include correlation tend to underestimate uncertainty or variability of results, sometimes dramatically so.

* 1. **Distributions used to Model Stochasticity**

Two distributions are used throughout the watershed model to describe uncertainty and variability: The normal distribution, and the Dirichlet distribution.

* + 1. **Normal Random Variables**

Most parameters are assumed to follow a normal distribution, and the input files therefore specify a mean value, and a standard deviation corresponding to each component of uncertainty / variability. In many instances, the parameters are assumed normal despite the restriction that the values must be positive (i.e. total area). For these, the “positive normal” distribution is used within the R code.

* + 1. **Dirichlet Random Variables**

Parameters that describe probabilities and/or fractions that, by definition, must add up to 100%, cannot be adequately described by a normal distribution. Distributions for univariate probabilities (for example, the probability of survival from fry to parr) are commonly described by the Beta distribution, which allows for values between 0 and 1. The multivariate generalization of the Beta distribution is the Dirichlet distribution. This allows probabilities to be assigned to a number of possible events, where all probabilities must add up to one. For example, within a land use classification such as “forest”, there is an assigned fraction of habitat types, all of which must add up to exactly 1. To model uncertainty or variability in these fractions, the Dirichlet distribution is used.

Typically, a Dirichlet distribution is specified using a vector of alpha parameters. For example, if **X** is an n-dimensional Dirichlet random variable, it is specified as:

X ~ Dirichlet(α1, α2, … , αn)

The mean and variance of each fraction or probability is:

(6.1)

(6.2)

A complete discussion of the Dirichlet is not included here. Many standard statistics texts, particularly those describing Bayesian statistics, provide complete descriptions.

For the purposes of the watershed model, having the user specify a vector of α values for each component of variation would be quite tedious, and for non statisticians, perhaps excessively esoteric. For these reason, an alternate, though somewhat restrictive, parameterization is used throughout this model for all Dirichlet variables (including Beta variables, which are merely two parameter Dirichlet random variables).

First, the mean probability or fraction for each outcome is specified. This should be fairly self explanatory for the user. For example, the following table shows a hypothetical set of values for the average fraction of various habitat types for the land use classification of “forest”:

**Table 6.1. Example Site Specification for Mean Fraction of Habitat Type within Land Use Classification “Forest”**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| LU\_Classification | pools | cascades | glides | Riffles | rapids |
| Forest | 0.6 | 0 | 0.25 | 0.1 | 0.05 |

Note that the sum of all the values must add up to 1.0. In order to add specification for the uncertainty or variability associated with each of the values, a single multiplier “N.alpha” is used for each component of variability. This multiplier is used to generate a set of alpha values describing the Dirichlet distribution:

αi = µi \* N.alpha (6.3)

For example, if N.alpha = 100 is used for the habitat types with the forest land use classification above, we get the following set of α values:

**Table 6.2. Dirichlet α values corresponding to Mean Values of Table 5.1 and “alpha.N” = 100**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| LU\_Classification | pools | cascades | glides | Riffles | rapids |
| Forest | 60 | 0 | 25 | 10 | 5 |

Thus we have multivariate random variable distributed, Dirichlet(60, 0, 25, 10, 5). From this, and equations 6.1 and 6.2, above, we see we get the same vector of means as specified in Table 6.1, and the standard deviation (square root of the variance) of each fraction is as follows:

**Table 6.3. Standard Deviation Values based on Dirichlet Distribution of Table 6.2**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| LU\_Classification | pools | cascades | glides | Riffles | rapids |
| Forest | 0.0049 | 0 | 0.0043 | 0.003 | 0.0022 |

Thus, with a set of mean proportions, and a single value for “alpha.N”, we can specify fully specify a Dirichlet distribution and calculate standard deviations for all proportions. Note that these are not independent from one another. This must be so to ensure all values in the row add to one in call cases.

Using alpha.N to specify variability is useful not only in that it allows the user to specify a single number for each component of variability (run to run, year to year, site to site, within site), but also in that in may be thought of as a value that has some practical meaning, especially when thinking of variability as describing uncertainty of a parameter***. The alpha.N value may be thought of as the number of samples from which the estimated means were obtained***. Thus, increasing the alpha.N value results in less uncertainty in the estimated means. As alpha.N values are increased, the standard deviations go down.

When specifying year to year, site to site, or within site variability, thinking of alpha.N in this way may not be as intuitive, as it’s not really describing an uncertainty – it’s describing actual, natural variability. However, it may still offer the user some insight into how to use alpha.N to describe variability: higher alpha.N values correspond to lower variabilities.

The user is NOT allowed to enter standard deviations directly and let the program calculate alpha.N or a vector of **α** values from the standard deviations; this precludes the user from entering input values that are not consistent with a valid Dirichlet distribution.

* 1. **Correlation Structure of Input Parameters**

This section describes how the values entered in the input file(s) are used to create correlation structures for each parameter used in the model. It is not critical that the user understand this section in order to run the model. However, understanding this section will enable the user to input meaningful, insightful variability estimates to the input files, as well as fully understand and appreciate the results.

As described in section 4, the input file allows users to specify mean and component variabilities for each parameter included in the model. It’s important to note that, by breaking the components of variability into run to run (uncertainty in parameters), year to year, site to site, and within site, appropriate correlation structures are induced in the data. These correlations are important as they may have significant impacts on the variability of model results.

* + 1. **Correlation Structure for Normally Distributed Variables**

For normally distributed random variables, inclusion of correlation structures are illustrated using mean land cover by site (Ak\_x\_Lqk) as an example. Ak\_x\_Lqk describes land cover by area for a given site. It could be entered in the input file as follows for a simulation with 7 defined land use classifications:

**Figure 6.2.1. Ak\_x\_Lqk**



This would specify, for example, that the average land cover of category forest for this site is 400,000 square meters. (Section 7 will detail specifics on entering these values into input files).

Random effects for run, year, site, and within site effects are included to describe the land cover for run r, year t, site k, and land use category q as follows:

Ak\_x\_Lqk[r, t, k, q] = µ Ak\_x\_Lqk +ar + b t(r) + c s(t, r) + e q(r,t,k) (6.4)

Where **a** is the random effect of run, **b** is the random effect of year (nested within run), is the random effect of site (nested within year and run), and **e** is within site random effect (nested within site, year, run). This structure will naturally induce positive correlations for land cover values within a site, sites within a year, and years within a run. One might expect such correlations to occur in nature, thus it is important that this correlation structure be enabled in the watershed model.

Implementation of this correlated variance structure in the R program is as follows. In order to generate correlated variability components, we first randomly generate probability quantiles from uniform[0,1] distributions that will serve as quantile probabilities, from which to select variance components. These probabilities are listed in the R code as follows, for simulation run r, at time t, site k, and land use classification q, the following quantile probabilities are defined:

r.unif[r] = runif(0,1)

T.unif[r, t] = runif(0,1)

s.unif[r, t, k] = runif(0,1)

Withins.unif[r,t,k,q] = runif(0,1)

For a given run, year, site, and land use classification, the land cover is assigned as follows:

Ak\_x\_Lqk[k, t,q, r] = Ak\_x\_Lqk.mu +

qnorm(r.unif[r], 0, σr) +

qnorm(t.unif[r,t], 0, σt) +

qnorm(s.unif[r,t,s], 0, σs) +

qnorm(Withins.unif[r,t,k,q], 0, σ) (6.5)

Note that all year / site / land use classifications (indexed by t,k, and q) share a common random effect for simulation, by virtue of the fact that “r.unif” is held constant across all t, k, and q within a simulation. Therefore, land cover values within a simulation run will be positively correlated with each other as compared to land cover across all simulations. Similarly land cover with a year will be positively correlated within a given run; and land cover within a site will be positively correlated within a year.

* + 1. **Correlation Structure for Dirichlet Random Variables**

For parameters modeled as Dirichlet random variables, a structure similar to that above, as described for normal random variables, is implemented. However, some extra complexity is encountered due to the fact that, for Dirichlet random variables, we cannot simply add, linearly, a random effect to each component, as that would not ensure the total probabilities always added to one (i.e this would not result in a valid Dirichlet distribution).

However, if, say, X ~ Dirichlet(α1, α2, … αn), we can take advantage of the fact that a random values from a Dirichlet distribution may generated by randomly drawing a value from a gamma(αj, 1), then assigning values to x1, x2, .. ,xn as follows:

(6.6)

(6.7)

where αj is the alpha value described in section 6.1.2, and j is indexed from 1 to the number of probabilities for fractions that must add up to one.

In order to generate correlated random Dirichlet variables, as was done for normally distributed random variables, the implementation strategy is illustrated via using the proportion of habitat type, by land use classification, as an example. The following table shows example input data for this scenario. Recall from section 5.1.2 how alpha values are generated for run to run, year to year, site to site, and within site variability components.

For example, the proportion of each habitat type, by land use classification, might be specified as follows:



Let M[r, k,q,j,t] be the proportion of habitat type j for run r, at site k, in land use category q, at time t.

First, values for a specific run, r, are drawn. This is done by randomly generating a vector of unif[0,1] variables of length j, where j, in this case, is the number of habitat types, and the sum of M across all j’s must be equal to one. J (capitalized) indicates the total number of habitat types.

r.unif = runif(J, 0, 1) generates a vector of J random uniform[0, 1] variables.

Define a temporary variable, y[r, k, q, j, t] as:

(6.8)

And define “Mr” (M for run r, for each year, site, land use type)

(6.9)

Note that, at this point, the vector r.unif is shared across all years and sites within a run. Thus, as with normal variables, we will generate a positive correlation for values of M within a run as compared to across all runs.

From the new values, we can generate a new “run specific” vector of **α** values, and repeat the above process to generate values for each year within a given run. This is done exactly as above, except that the vector of uniform variables, t.unif, is now randomly re-assigned for each year within each run, and M is allowed to vary year to year (noting that the starting vector of **α** values is common to all years within a run). M values within a year will be positively correlated with each other.

Similarly, to generate site specific M values within a year, we again use updated year (nested within run) specific vectors for **α**, and generate site specific M values. A vector of J uniform(0,1) random variables, t.unif, is now generated for each site within each year within each run, and the above process is repeated for each site. In this manner, positively correlated values of M will be generated within each site, as compared to across all sites within a year.

Finally, within site variation is added by, again, generated site(year(run)) specific **α** vectors and generating a random vector WithinS.unif of J uniform(0,1) random variables. WithinS.unif is re-randomized for each land use category within each site within each year and within each run.

At this point we have all required values of M, positively correlated within run, within year(run), within site(year, run), and within land use category(site, year, run). The magnitude of the correlations is driven by the variability component at each level of the hierarchy. If the user has reasonable estimates for variability components, he or she need not generate estimates of correlation. As described above, they will implicitly be generated by the watershed model.

1. **Including Time Based Changes for Input Parameters**
   1. **Continuous Time Based Trends**

Specifying mean trend shifts for parameters is done by utilizing cells on the input files labeled “future mean” and “rate parameter (lambda)”. For all mean shifts over time, the mean shift is modeled using an exponential growth / decay model of the form:

(7.1)

Where is the value of the parameter mean at time t, is the parameter mean at time 0, is the target (long term future mean value) of the parameter, and is the rate parameter which controls the rate at which the parameter mean changes from to .

Using an exponential function, as opposed to a simple linear model for time trends, provides a flexible approach for describing how parameters change over time. Many natural processes may be modeled in this manner. For example, a step change in management practice of a site within a watershed may lead to a change in land use categorization that initially changes fast, but over time asymptotically levels off to a new long term mean.

**Figure 7.1. Examples of Time Trends that can be specified with exponential growth / decay model**



Within the framework of equation 7.1, a nearly linear change in means can be described by the above equation by using only the “linear” portion of an exponential decay curve, so that, within the time period being modeled, the mean only moves a small fraction of the way toward the “target” value. Figure 7 .1 demonstrates how varying µt and λ can produce a range of curves for the parameter mean over time. This example demonstrates various trends of increasing Land Cover; of course, decreasing trends can be implemented just as easily.

Note that, in the process of running the Monte-Carlo simulations, the within run quantile probabilities will carry over from the initial mean values to the “target” mean values. For example, if for a given run, the initial mean land cover for a given land use category was randomly assigned to the 78th percentile in the specified normal distribution, the target value in the trend calculation would also be at the 78th percentile. (The standard deviation used is the same for the initial and target values). This will preserve the integrity of the Monte-Carlo simulation. Similarly, all variability components and correlations will be preserved as temporal trends are imposed.

* 1. **Step Functions for Changes in Parameter Means**

Discreet step function changes in parameters, or discreet changes in time based trends, can also be user specified. This is done by creating unique input files (see section 8) for each point in time at which the user requires a step function change in any parameter or parameters. (This can include step function changes in parameter means, variabilities, future target values, or rate parameters). One or more step function changes can be entered, with a maximum number of step changes limited only by the number of years included in the simulation. Whether discreet changes in one or many inputs are to be specified, the entire input file will need to be specified for each step. The easiest way to do this is to create an initial input file, copy it to a new input file, and change only those cells for which step changes are to be made.

The header file is where the user specifies which input files are to be used for which time step ranges. The user may specify as many input files as desired, as long as the name of each input file and the ranges of time to use each are fully specified in the header file. (See section 8 for full details on specifying a header file).

**Figure 7.2. Example Header File to specify discreet changes in input parameter data**





For example, figure 7.2 shows an example header file that specifies a step function change in inputs one site. “Site A” uses inuts from file “Site1\_T1.csv” from year 1 to year 50, then switches to inputs specified in “Site1\_T2.csv”. Note that all other sites use the same input files at year 1 and year 50. However, since a time step occurs in at least one site, the input file names must be specified at t=50 for all sites; in this case the input files used at t=50 are simply the same files as used at t=1 for sites B through K. Details in the header file specification and input files specifications are in section 7.

Note that, in the process of running the Monte-Carlo simulations, the within run quantile probabilities will carry over across discrete jumps in the data inputs. For example, if for a given run, the land cover for a given land use category was randomly assigned to the 78th percentile it the specified normal distribution, it would continue to use the 78th percentile across all discrete step function changes. The distribution parameters may change, therefore the actual parameter values may change, but the quantiles will be consistent within a run in order to preserve the integrity of the Monte-Carlo simulation and all other variability components.

* 1. **Combining Mean Trends and Step Functions for Changes in Parameter Means**

The user may specify trends in the mean, and step function changes, within the same simulation. However, care must be taken when doing this, as parameter means will always make discreet jumps at the change in input files. If, say, a trend change in parameter A overlaps a discrete change in parameter B, the user will want to estimate ahead of time the mean for parameter B at the time of the discrete change. For the input data file representing the time of the discrete change, the estimate of mean of parameter A at this time should be specified as the initial value.

Using a combination time based trends within a user input file, and multiple input files for discreet step changes in either input parameters or their associated time based trends, the user has sufficient tools to stitch together patterns of input parameter time trends that reasonably trace any time based function desired.

* 1. **Stochasticity and Mean Trends**

The stochasticity described above is applied in conjunction with discrete step function changes and / or temporal trends in parameter means. The effect of this is that trends of the mean values over time are not smooth transitions, but rather better characterized as random variation about a central tendency defined by the future target and rate parameters. Figure 6.3 is an example showing land cover for forest, in site 1 of a given simulation, where the initial mean value is 400000, the target value is 200000, and the rate parameter lambda is set at .25. At t=30 years, a step function changes was specified such that the new starting mean value is 200,000, the new target value is 300,000, and the new rate parameter is 01.

**Figure 7.3. Example of Temporal Variation with Time Trend and Step Function Change (at Year 30) for Land Area (Ak\_x\_Lqk)**



1. **Entering Simulation Data and Parameter Estimates into the Input Files**

Inputs parameters, site information, etc., are specified by filling out a set of .csv files. The R program will read data from these .csv files, run the simulation, and output results both within the R workspace and to a set of output .csv files. Because of the complexity involved in specifying model parameters for multiple sites for multiple life stages, habitat types, etc.; and the complexity in including modeled stochasticity to account for uncertainty in parameters as well as temporal, site to site, and within site variability; and to allow for specification of parameter changes over time, either as trends or step functions; the input files are necessarily complex. Much effort has been made to make the data input process as simple and manageable as possible; however, the bulk of the user’s time using this model will be devoted to fully and accurately entering data into the various input files.

Also, please note that these files are read from R programs. Unlike working within, say, a spreadsheet within Excel, ***R does not recognize cell location changes*** if a row or column has been added, or a cell has been moved. It is critical that the location of all cells in the input files remain exactly as provided. The user is advised to make backup copies of all user input files included with the model. The user will then start with a copy of the input files provided, and edit them per his or her requirements.

Also note that, in general, **all cells in all input files must contain values, even if the values are zeroes**. The watershed model code is not sophisticated enough to recognize blanks as zeroes, nor does it conduct any check on the integrity of the input files. Note also that certain sets of inputs have constraints (for example, a set of probability inputs must add up to one within a row) that must be followed correctly for the code to work correctly; but there is no check built into the code for these constraints. User beware!

* 1. **Entering Information into “header.csv”**

The file “Watershed\_Header\_File.csv” contains basic information such as the number of sites being modeled, the number of time steps to model, and the file names of all other input files. This file must be named “Watershed\_Header\_File.csv” to be recognized by the R code.

Figure 8.1. shows the input and information contained within the header file. (Note that the file formatting is here for reference only. CSV files do not enable file formatting. A formatted, Excel version of this file is available as a template, but the file must be saved as a .csv file to use with the R-program). Input cells are shown in blue.

**Figure 8.1. Header File Input: Simulation Details**







Data to be entered includes: the number of sites, or habitat locations within the watershed, to be included in the model. A maximum of 10 sites may be included. The number of input files *per site* is greater than 1 if there are step function changes included in the simulation. In other words, if at some point in the simulation the user wishes to switch input files from one set of files to a second set of files (with a “set” being one input .csv file per site), then the number of input files per site is one plus the number of step changes. (More details below). The number of land use classes and the number of habitat types must be specified (see sections 7.2.1 and 7.2.2). The number of time steps indicates the years for which to run a simulation. The number of Monte-Carlo simulations indicates how many iterations of the entire simulation to conduct, to be used if run-run stochasticity is specified (with a maximum of 20 or 30 monte-carlo simulations generally being sufficient to assess uncertainty in results due to uncertainty of the inputs. Specifying more than 30 may cause the code to take many, many hours to complete a set of simulations).

The MC1, MC2, MC3, and MC4 inputs are toggles to “turn on” or “turn off” different aspects of stochasticity in the simulation. An input of “0” tells the code NOT to include that element of stochasticity, while and input of “1” tells the code to include stochasticity for that portion of the model. “Turning off” stochasticity in this manner has the same effect as setting all variability parameters to values indicating zero variability; the MC1, MC2, MC3, and MC4 toggles are provided as shortcuts to allow the user to avoid having to recreate input files when examining the relative effects of different elements of stochasticity.

The remainder of the required header file information pertains to additional input file names and simulation times at which various parameter input files should be used. The first column to be user specified is the “Site Name” column. This column should include one row for every site to be modeled; thus the number of rows specified should be equal to the “Number of Sites (K)” input, specified above. Any rows specified, beyond the number specified above, will be ignored. In this example, 10 sites are specified. The names in the “Site Name” column will be used when creating site-level plots and output files. Note that “\”, “/”, and other special characters should not be used in site names, as they will cause errors when site names are used in output file names.

For the next column, the user must specify filenames for .csv files containing initial values, for all sites included in the model. Detail on generating these files is provided in section 7.4. Each site must have a file specified. However, the user may use the same initial values file for multiple sites. In this example, sites 5 through 10 all share the same initial values, thus a single file is used in all cases. (This might be used, for example, if these starts start at time 0 with no fish, and change over time due to migration, etc. Or the user may simply “seed” all sites with the same number of fish, and allow the model to run to some steady state; in which case using a single initial values file across all sites would be appropriate).

For the remaining columns to the right, one column must be specified for each “Number of Input Files per Site” specified as in Figure 7.1.1. In this example, two columns are required. At the top of each column, the user must specify “Input File Times”. The initial input time MUST equal 1, as the simulation must start at year 1. If multiple input files per site are used (to specify step changes in input parameters), then the user must specify at which points in the simulation the step changes are to occur. In this example, the first set of input files is used at year 1. The second set of input files is to be used by the simulation beginning at year 50, as specified by the upper-right cell in Figure 7.1.2., for this example. The final time step (t=100 in this example) does NOT need to be specified here, as no input file change occurs at the final time step. If additional step changes were to be used, the total number of input files would be specified as shown in 7.1.2, and an additional column for each additional input file would be added to the right of the columns as shown in Figure 7.1.2. There is no limit on the number of input file changes, other than no more than 1 file per year of simulation can be specified.

Below the input file times, the filenames for the input parameters for each site/input file time combination must be specified. In this example, the input file parameters for site 1, at year 1, are specified in the file “Site1\_T1.csv”. The input file parameters for site 1, beginning at year 50, are specified in “Site1\_T2.csv”. The inputs parameters will move, in a step function, from those specified in “Site1\_T1.csv” to “Site2\_T2.csv” at year 50 of the simulation. For this example, no other site contains a step change in the input parameters. However, since we have a step function for one site, input files must be specified for all sites at all step-function time changes. However, a unique input file is not required for each time step for sites 2 through 9. Instead, the same file name is specified for in both columns, indicating that the same input file will be used before and after year 50 for those sites.

Finally, the name of the cross-site migration matrix .csv file(s) must be specified. Just as for the parameter input files, one file name must be specified at each of the input file times. A step function change in cross site migration probabilities can be modeled b switching from one file to another file, just as with the site level parameter input files. If no step changes in migration are to be incorporated, simply use the same file name in all columns, as is shown in this example. Details on creating the cross site migration input file are provided in section 7.3.

* 1. **Entering Input Data into the Watershed Data File(s)**

The bulk of the data for the Watershed model is entered in .csv files, with filenames as specified in the “Watershed\_Header\_File.csv” file (see figure 7.1.2). If no step function changes in input parameters are specified, the user will create one input file for each site included in the model. Additional input files will be added for each time step where a step function change or changes are made to the input parameters for a given site. All parameter input files must have exactly the same format; and, as in the “Watershed\_Header\_File.csv” file, cells must not be moved, rows or columns may not be added, etc.

Users may model from 1 to 10 sites concurrently. Rather than create an input file from scratch, a user will want to begin with an existing .csv input file, and modify it for the site being modeled. Alternately, the user may use an Excel file template (which includes some formatting and extra annotation which may be useful to the new user). This Excel file can be filled out completely, and then saved as a .csv file under a filename specified in the header file.

Below is a description of each set of inputs that must be filled out for each site. Figures include formatting as will be found in the Excel templates. As the user gains experience with the model, he or she may prefer work directly with .csv files. The .csv files are exactly the same as the Excel files, except that formatting (cell colors, font, column widths) are lost.

As mentioned early, creating the site level parameter input files will likely comprise the bulk of the work when using the watershed model.

The file “Input File Template.xlsx” should be consulted while reading this section. Within this file, cells in blue are those that must be user specified. Cells in orange are cells that the user may specify for reference purposes, but are not actually read by the model. These reference cells may include names of land use classifications, names of habitat types, and comments. The color coded Excel version of the input parameter file is used in this document to aid in illustration of the inputs.

**Land Cover by Area (Ak\_x\_Lqk)**

The first input table to be filled out is for site level **Land cover by area**  or **Ak\_x\_Lqk**, (as described by Table 2-2 of QCI, 2006). This input parameter is modeled as a normally distributed random variable. Data for land cover by site is entered in the top most table of each site data input file. An example is shown by Figure 7.2.1.

**Figure 7.2.1. Land Cover by Land Use Classification for each Site**



Table 7.2.1 includes user data for land cover by area by land use classification In the first column, the user specifies the mean value, and in the next four columns, the user must enter run to run standard deviation (uncertainty for the Monte-Carlo simulation), followed by year to year, site to site, and within site standard deviations for Ak\_x\_Lqk. The final two columns allow the user to specify a future target for Ak\_x\_Lqk by land use classification, and a rate parameter that governs the rate of change from the initial to the target value (see section 6.1**). If the user does not wish to specify a trend change in mean Ak\_x\_Lqk over time, he or she can either specify a rate parameter of 0, and/or specify the future Ak\_x\_Lqk target mean as being equal to the initial mean parameter value.** (This will hold true all input file parameters). Note that, if no stochasticity is desired, the user must still enter zeroes in the sigma.x columns.

The units on area may be selected by the user. However, the units on area MUST be consistent with the units for capacity, discussed below. For example, if Ak\_x\_Lqk is specified in square meters, then the capacity values must be specified in fish per square meter.

**Fraction of Habitat Type within Each Land Use Classification (M)**

Figure 7.2.2 shows an example of user inputs required to specify the fraction of habitat type within each land use classification (see Table 2.3 of ref QCI 2006). These inputs, being a series of proportions that must add to 100% within in land use classification, are modeled as Dirichlet random variables. The first sets of columns are used to specify the mean proportion of each habitat listed. Note that the user may include up to 12 habitat types within each land use classification, and up to 12 land use classifications. If fewer than 12 habitat types and/or land use classifications are used, the unused rows and columns should contain zeros. (The zeros are ignored, but the functions used to read the .csv files from R may need a zero in these files to properly read and interpret the data).

As described in section 5.1, values of “alpha” are used to specify precision for series of fractions in each row of the table shown in Figure 7.2.2. N.alpha values are specified for run to run variability (Monte-Carlo uncertainty) as well as year to year, site to site, and within site variability estimates. Unlike standard deviations used for normal random variables, increased N.alpha values specify increased precision rather than increased variability. A “zero variability” default value for the Alpha values is infinity. Using 9999 as a default, for modeling purposes, provides a “zero variability” default value (Note, to get a true “zero” variability would require an infinite value for Alpha. Using 9999 or any very large number is sufficient for modeling purposes).

The right most columns specify the future target values and change rate parameters. Again, refer to section 6.1 for details on how to use this. As with normally distributed random variables, the future target values should be entered whether a trend is to be modeled or not. If no trend is specified, the future target should be set equal to the initial mean value. (Note that columns have been broken apart for visualization here. All tables shown in the figure are, in the input files, on the same sets of rows).

**Figure 8.2.2. Fraction of Habitat Type within Each Land Use Classification (M)**





**Capacity by Life Stage and Habitat Type**

Figure 8.2.3 shows example input for capacity by life stage / habitat type. These is modeled as a normally distributed random variables. This table includes columns for multiple life stages, thus has significantly more columns than some input tables. Conventions for entering means, standard deviations, future targets, and rate parameters are consistent with other parameters modeled as normally distributed random variables. **The units for capacity are fish per area, and the units on area MUST be the same as the units on the Ak\_x\_Lqk values specified in the first table.** For example, if Ak\_x\_Lqk is specified in square meters, then the capacity values must be specified in fish per square meter. (Note that columns have been broken apart for visualization here. All tables shown in the figure are, in the input files, on the same sets of rows).

**Figure 8.2.3. Capacity by Life Stage and Habitat Type**





**Productivity Scalar by Life Stage and Land Use Category**

Figure 8.2.4 shows and example of user inputs for Productivity Scalar (see Table 2-2 in QCI 2006). This is modeled as a normally distributed random variable. Conventions for entering means, standard deviations, future targets, and rate parameters are consistent with other parameters modeled as normally distributed random variables. . (Note that columns have been broken apart for visualization here. All tables shown in the figure are, in the input files, on the same sets of rows).

**Figure 8.2.4. Productivity Scalar by Life Stage and Land Use Category**





**Survival Probability by Life Stage**

Example user inputs for Probability of Survival by life stage are Figure 8.2.5. For each life stage within each site, these are modeled as Dirichlet random variables (and since the outcome is either death or survival, the Dirichlet variable is equivalent to a 2 parameter Beta random variable). Note that a row for the One-Plus life stage is grayed out for these inputs. One-Plus inputs, due to the added complexity required to enable modeling of steelhead, are included as a separate input table. This row is not used, and should not be filled out at this stage of the process (regardless of whether Steelhead or non-steelhead species are modeled). However, do leave the default “-99” values in these cells. (The R-functions that read that data are not robust against reading in mixtures of numeric and non-numeric data. Non-numeric characters may cause errors). Conventions for entering means, standard deviations, future targets, and rate parameters are consistent with other parameters modeled as Dirichlet distributed random variables.

**Figure 8.2.5. Survival Probability by Life Stage**



**Figure 8.2.5.2. Subyearling Inputs for Chinook**



**One-Plus Inputs**

As mentioned above, enabling Steelhead modeling requires added complexity, specifically implemented in the tracking of One-Plus. This added complexity is generalized, and used whether steelhead or non-steelhead species are being modeled. The user input process will be the same for both.

Figure 8.2.6 shows an example set of inputs for One-Plus. Note that the inputs allow for fish to remain as One-Plus for up to 10 years. Non steelhead species will smolt immediately (i.e. at year 1); therefore, rows beyond the first row will not be applicable. To specify non-steelhead smolt behavior, “N5.Psmolt” for year 1 would be set to 1. (Note that columns have been broken apart for visualization here. All tables shown in the figure are, in the input files, on the same sets of rows).

The first sets of inputs are probability of survival by year as One-Plus. This enables the user to specify different survival probabilities for older One-Plus (or resident rainbows) and for younger or new One-Plus. Survival, as in other life stages, is modeled as a Dirichlet (or more simply, a Beta) random variable. Conventions for entering means, standard deviations, future targets, and rate parameters are consistent with other parameters modeled as Dirichlet distributed random variables.

The next set of inputs in this table require specification of the probabilities that a given fish either smolt (head to the next life stage), stay another year as a One-Plus, or return to spawn as a “resident rainbow” that never migrated to the ocean. This set of parameters is modeled as a Dirichlet random variable with three potential outcomes (smolt, stay, spawn). These three probabilities, in each row, must add to 1.0.

A non-steelhead species will arrive as a One-Plus, and smolt (attempt to migrate to the ocean) with 100% probability in the next year. A steelhead will have less than 100% probability of smolting each year. Note that the probabilities in each row represent the probability of smolting at that year, given that the fish has not previously smolted; i.e. this is a conditional probability. As such, the probabilities in each *column* do not add to 1. Conventions for entering means, standard deviations, future targets, and rate parameters are consistent with other parameters modeled as Dirichlet distributed random variables.

**Figure 8.2.6. Pre-Smolt Inputs (note: not all inputs are shown. Male smolt/spawn/stay inputs are equivalent to female inputs; target and rate parameters not shown).**







Also included in the inputs shown by Figure 8.2.6 are capacity scalar values (see section 4.6). These are factors used when modeling steelhead (and should simply be set to 1.0 for non-steelhead species). A fish that has remained as a One-Plus for two or more years is presumably larger than a “new” One-Plus. The capacity scalar is used to quantify that additional size so that, in the Beverton-Holt algorithm, the capacity of the site (maximum number of fish per meter squared) is reduced by the weighted average of the capacity scalar (weighed by the number of fish of each age). This ensures that the algorithm recognizes that older, larger One-Plus require more area than younger One-Plus. (Note: this and relative survival probabilities are used to determine how many of each age survive, year to year, given that fish of multiple ages are competing for the same space. See section 4.5). The capacity scalar is modeled as a normally distributed random variable. Conventions for entering means, standard deviations, future targets, and rate parameters are consistent with other parameters modeled as Normal random variables.

Finally, fecundity for fish that choose to spawn directly from the One-Plus life stage, rather than migrate to the ocean, by One-Plus age, are given in the final column. It may be expected that these fecundity values are significantly less for these “resident rainbows” than they are for steelhead returning from the ocean. Also note that these fecundity values are on a per fish basis, rather than a per female fish basis. Unlike nearly all other parameters, uncertainty and variability inputs are not included for fecundity values. (These variabilities may be added in the future if desired by users).

**Adult Salmon: Spawn Probability and Ocean Capacity**

Figure 8.2.7. shows example inputs for parameter describing adult salmonids – those fish that have migrated to the ocean. For these parameters, mean values, standard deviations (or N.alpha precision estimates if Dirichlet variables), future targets, and rate parameters must be specified for fish, by ocean age:

* Probability of returning to Spawn (modeled as a Dirichlet (Beta) random variable);
* Ocean carrying capacity by ocean age, modeled as a normally distributed random variable
* All variables follow the same conventions for specifying means, variability, future targets, and rate parameters as described for previous parameters.

**Figure 8.2.7. Adult Salmon: Spawn Probability, Fecundity of Spawners, and Ocean Capacity**





**Fractional Watershed Site Area due to Seasonal Flow Changes**

Figure 8.2.8 shows example inputs for “Fractional Watershed Site Area due to Seasonal Flow Changes”. These values account for the fact that, from season to season, the surface area of water within a site will change, and the amount of this change will vary by habitat type. For example, in the summer months, reduced water flows may reduce surface area of rapids by 75% for a given stream site. The surface area within a deep pool may be reduced only slightly during this time. Seasonal changes, as are other variables, are modeled such that year to year, site to site, and within site variability correlation structures will give rise to correlations. This may be especially important to consider for this variable, as seasonal changes will undoubtedly be correlated across sites within a given year due to weather patterns affecting entire watersheds.

**Figure 8.2.8. Fractional Watershed Site Area due to Seasonal Flow Changes**















All variables follow the same conventions for specifying means, variability, future targets, and rate parameters as described for previous normally distributed parameters.

**Fraction of Fish Harvested**

Next, for each site, the user must specify a fraction of total fish harvested. No distinction here is made between commercial and recreational harvests. This parameter is modeled as a normally distributed random variable. (Note, this could be modeled as a Dirichlet as well, and give that it’s a fraction, one might argue this is a more appropriate model. However, the variation around this is expected to be small as this is a highly managed and controlled process. As such, a normal distribution should suffice). Fractional harvest specifications follow the same conventions for specifying means, variability, future targets, and rate parameters as described for previous parameters.

**Figure 8.2.9. Fraction of Fish Harvested**



Note that figure 7.2.9 includes inputs for both wild and hatchery harvest rates, as they are generally managed to different targets. Hatchery fish, in this case, includes only hatchery bred fish (H1). Off-spring of hatchery fish that bred in the wild, or various combinations of offspring from wild and hatchery fish, are considered “wild” from a harvest viewpoint, since they would be identified as wild by fishermen.

**Hatchery Introductions**

Figure 7.2.10 shows the user inputs related to hatchery fish introductions. The top row simply lists the number of hatchery fish per year introduced at each life stage, for a given site. Fish can be introduced as eggs, fry, One-Plus, or smolts. Fish can be introduced at multiple life stages during a single simulation.

Note that, in the current version of the model, stochasticity and temporal trend changes are not enabled for hatchery introductions. Step function changes can be modeled as in all other parameters.

**Figure 8.2.10. Hatchery Introductions**



**Hatchery Effects on Survival: Relative Survival Probability**

The next sets of inputs require user specification of the relative survival probabilities of natural, or “wild”, fish; hatchery fish; and all genetic combinations of hatchery and wild fish up to three generations removed from hatchery origin.

Finally, the relative fecundity for fish of a given ancestry is user input on the right hand column of this table. This, as described in section 4.12, is a scalar applied at the egg stage for eggs of the given genetic category.

**Figure 8.2.11: Hatchery Effects on Survival: Relative Survival Probability**



Note that, in the current version of the model, stochasticity and temporal trend changes are not enabled for hatchery effects on relative survival probability. Step function changes can be modeled as in all other parameters.

**Fecundity of Female Spawners**

In the next set of required inputs, the user must enter estimates for fecundity of female spawners. As this is thought be vary by genetic sub-species type, and certainly varies by age of the returning female fish, the user must specify estimates for all age/genetic sub-species type combinations, as shown in Figure 7.2.12.

**Figure 8.2.12 Female Fecundity by Years as Adult and Genetic Sub-Species Type (For Hatchery Effects)**



Note that, in the current version of the model, stochasticity and temporal trend changes are not enabled for female fecundity. Step function changes can be modeled as in all other parameters.

**Proportion of Females in Spawning Population**

Finally, the proportion of female spawners, relative to all spawners (male and female), must be specified. As for fecundity, the model enables users to specify the proportions uniquely for different ages of returning spawners, genetic sub-species types, and all combinations thereof. Figure 7.2.13 shows example inputs for these parameters.

**Figure 8.2.13 Fraction of Females in Returning Spawners by Genetic Subtype**



Note that, in the current version of the model, stochasticity and temporal trend changes are not enabled for the proportion of females. Step function changes can be modeled as in all other parameters.

* 1. **Cross Site Migration Matrices**

The cross site migration probabilities, as discussed in section 4.11, are entered on the cross site migration matrix input file(s), with filename(s) specified in the header file “Watershed\_Header\_File.csv”. The user must specify the probability that, for a given life stage, a fish will migrate from one site to another site. The model includes cross site migration potential for life stages of fry, parr, One-Plus, and returning spawners. If steelhead fish are being modeled, the One-Plus life stage can cover a range of fish ages, since steelhead may stay in this life stage for a variable amount of time. For simplicity, the model assumes that any cross site migration happens uniformly across age distributions at this stage.

As probabilities, the cross site migration probabilities are modeled as Dirichlet random variables. As with other Dirichlet random variables, run to run variability, as well as temporal variability, site to site variability, and within site variability is specified by an alpha parameter (as described in section 5.2.2).

Note that the sum of each row, including “migration” from a site to itself*,* ***must add up to one****.* Fish are neither created nor removed at this stage, merely redistributed across sites. Columns do not need to add up to one.

**Figure 8.3.1. Cross Site Migration Matrix for Fry**





If the user wishes to assume no cross site migration, he or she must input 1 along the diagonal, and zero everywhere else.

Figure 7.1.1 shows an example cross-site migration matrix for the fry life stage. Note that the input file is formatted to allow for up to 10 sites to be modeled in a single simulation. The user need only fill out rows and columns up to the total number of sites included in the model. Rows and columns higher than the number of sites are ignored.

* 1. **Entering Initial Values into Initial Values Files**

Because fish populations at life stages within the model are functions of the populations at the prior time step, populations at time 1 must be specified in the model. This is done by entering data into initial values files. One file must be present for each site modeled, and the names of these files are entered in the file “Watershed\_Header\_File.csv” (see section 7.1). In this file the user must specify initial (time 1) population numbers for each life stage, and each genetic subspecies type (wild, hatchery, all combinations thereof) at each site.

Figure 7.4.1 shows an example input file / worksheet for on sites. There is one row of column headers and one additional row for each of the 11 genetic subspecies varieties being modeled. The first column simply lists the site name (repeated for all 11 rows of data), and the 2nd column lists all the genetic subtypes. These first two columns are not read by the code, but should be included for reference. The remainder of the column contain initial population values for all life stages, including all possible ages of adults (ocean age 1 through ocean age 10) and all possible ages for One-Plus (1 through 10). Note that it is acceptable to enter zeros in any column or row. In many cases, the user will want enter “seed data” in only one cell and let the model run to some steady state.

In addition, the user may take the site-level results of any previous simulation, and use these results as the user inputs to follow up modeling runs (see section 7.4.1). T

Note that the One-Plus column is, in actuality, specified by a series of One-Plus inputs by years as One-Plus (columns X through AG). Thus, the vales entered in column K have no effect on the model. These “unused” columns are included in the input so that the input file can have the exact same format as the output file, making it easier to use the output data from a prior model as the input data to a new model.

Also note that the input data may contain fractional numbers of fish at each life stage. Obviously fish occur naturally in integer values; the model does not, however, distinguish between whole fish and fractional fish.

**Figure 8.4.1. Initial\_Values.xlsx example**





The user will likely not have information or want to enter initial quantities values for all life stages. Initially, the user may want to only specify an initial number of spawning fish per site at any chosen life stage or stages. He or she may then simply let the model run through many years of the simulation (> 30 years is recommended) until a steady state population is reached for fish populations at each site. Note that there are instances where the steady state solution is dependent on the initial values. Therefore, a somewhat reasonable, educated guess at initial values for at least one life stage is recommended. (An obvious example is where zero initial fish are specified at all life stages, in which case the “steady state” solution would remain zero fish at all times, as the modeled population would be extinct).

**8.4.1. Using Previous Results as Initial Values**

Another method for entering initial population numbers is to use the output from a previous run of the model. Upon the completion a simulation, a file is created for each site which contains the final population numbers at each life stage, for each genetic sub-type. These files are stored in a folder called “Output Files”, and are given file names of the form “TFinal [site name].csv”. These output files are in exactly the same format as the input files, and may be used directly as input files for a follow up run. To use these as input files, simply move these files into the working directory, change the file names to input file names (or specify the names in “Watershed\_Header\_File.csv” to match the output file names), and run the code.

1. **Running the Model in R**

Once the input files are prepared, as detailed in section 7, the user is ready to run the actual R script that performs the simulation. Note than since R is a command line programming environment, there cannot be a pre-compiled executable file. Instead, in order to run the program, the user must open R and run the main script manually. However, the user does NOT need to open the accompanying file containing additional R-functions. These are called automatically by the main script.

All input files and R-scripts must be located in a single folder on the user’s pc prior to running the program.

**Step 1: Open R and set the working directory**

Open R. Change the working directory to the folder where the R scripts and input files are located. This is done, with the R console as the active window, by clicking “File”, “Change Dir”, and navigating to the folder where the files are stored.

**Step 2: Open R-script “Watershed.R**”

Open the R script “Watershed.R” by selecting “File”, “Open Script”, and selecting “Watershed.R”. This is the main R script that calls all other functions required to run the Watershed model analysis.

**Step 3: Run the Script**

To run the entire program, from the “edit” menu, select “run all”. (A shortcut method is to use <ctrl-a> to select all, followed by <ctrl-r> to run the entire selection.) Depending on the number of sites, the number of Monte Carlo simulations, and the choices included for stochasticity, the amount of time required for the analysis will vary considerably. After reading the input files, the Monte Carlo / Beverton Holt simulation proceeds, with a message in the output window indicating the count of completed Monte Carlo iterations.

None of the other R files included in the model need to be opened by the user. The Watershed.R script will load each of the functions contained within these files. The user may wish to open these files only if he or she wishes to examine the code and/or modify the code in some way.

1. **Model Output**

Output for this model comes in two forms: output files, written as .csv files, and output plots written as files and displayed in the R-console gui. Output files are written to a folder called “Output Files”, which will be a sub-folder of the working directly. Note that**, when running the program, output data files and output plot files will overwrite existing files.**

**10.1.1. Fish Population by Site, and Totaled over All Sites**

Files containing the average and standard deviation (across all MC iterations) of the number of fish, at each life stage, at each year of the simulation, for each genetic sub-type, are generated for each site. Additionally, a file is created with the same numbers totaled over all sites. These population numbers are also broken out be genetic subspecies type. These files are named “Results\_[site name].csv”. One file per site modeled will be included. A file called “Results\_All\_Sites\_combined.csv” will be created showing the population totals across all sites.

The first couple of columns are simply an index column, and a column showing the time step of the simulation (these columns are duplicates of each other).

The next 17 columns in this output file list the number of fish (averaged over all Monte-Carlo iterations) at each life stage (Adult Escapement, egg, fry, … through all 10 modeled years of Adult Salmon). All One-Plus sub-life-stages are summed into a single column for the One-Plus life stage.

Note that the column names for these columns include “mean”, which indicates that these values are mean values (averaged across all monte-carlo simulations; even if only 1 run is done, with no stochasticity applied).

These column names also include “natural” in the column names. This indicates that the columns are sums of the genetic sub-type “natural”, indicating that these fish are NOT hatchery fish, and that they do not have any hatchery ancestry in the previous two generations of their ancestry. (“Natural” is used here instead of “Wild”, as “Wild” is, by convention, generally used to describe any fish not bred in a Hatchery – “H1” fish by this models convention – while “Natural” indicates, in this case, no hatchery ancestors in the preceding generations).

The next set of columns again lists average fish populations over all life stages, except these columns indicate Hatchery origin fish, and thus have “H1” included in the column names. Additional sets of 17 columns provide fish numbers for all other genetic sub-types (see table 4.12.1 for details).

Columns to the right of all the mean fish population columns, starting at the 190th column (column GH if opened in Excel) are used to provide the standard deviation of all fish numbers, by time, life stage, and genetic subtype. The standard deviation is applied across all Monte-Carlo iteration simulated. If only 1 Monte-Carlo iteration is specified, a value of NA will appear in these columns, as standard deviations cannot be estimated from a single data point.

**10.1.2 Smolts by Time and Genetic Sub-Type**

A file containing the number of smolts (fish leaving the watershed, attempting to migrate to the ocean), by genetic sub-type, at each life stage, are also generated for each site. Data include average and standard deviations (across all MC iterations). These files are named “Smolts\_by\_Subpop[site name].csv”. This information is duplicated in the “Results[ ].csv” files. It is output separately here as a convenience to the user.

**10.1.3. Spawners by Time and Genetic Subtype**

Similarly, files containing the number of spawners by genetic subpopulation are written, in the same format as the files outputting the number of smolts. These files are named “Spawners\_by\_Subpop[site.name].csv”. The information in the spawners output is not duplicated in the “Results[…].csv” files. Adult escapement (fish attempting to spawn that escape the fisheries) is list in the “results” files. “Spawners”, on the other hand, is a measure of that portion of adult escapement that survives to spawn (i.e. does not die of disease, exhaustion, etc; as calculated based on age and genetic-subtype adjusted survival probabilities) PLUS (for steelhead simulations) and “resident rainbow” spawners calculated for a given spawn year. Generally, the number of spawners is less than escapement, unless the number of resident rainbows exceeds the number of adult escapees not surviving to spawn.

**10.1.4. Population Values to Final Time Step**

Finally, files containing average (across all MC iterations) population values at the final time step, for each life stage and subspecies type, are created for all sites. These files can be used as initial values files for follow-on simulations, as described in section 7.4.1.

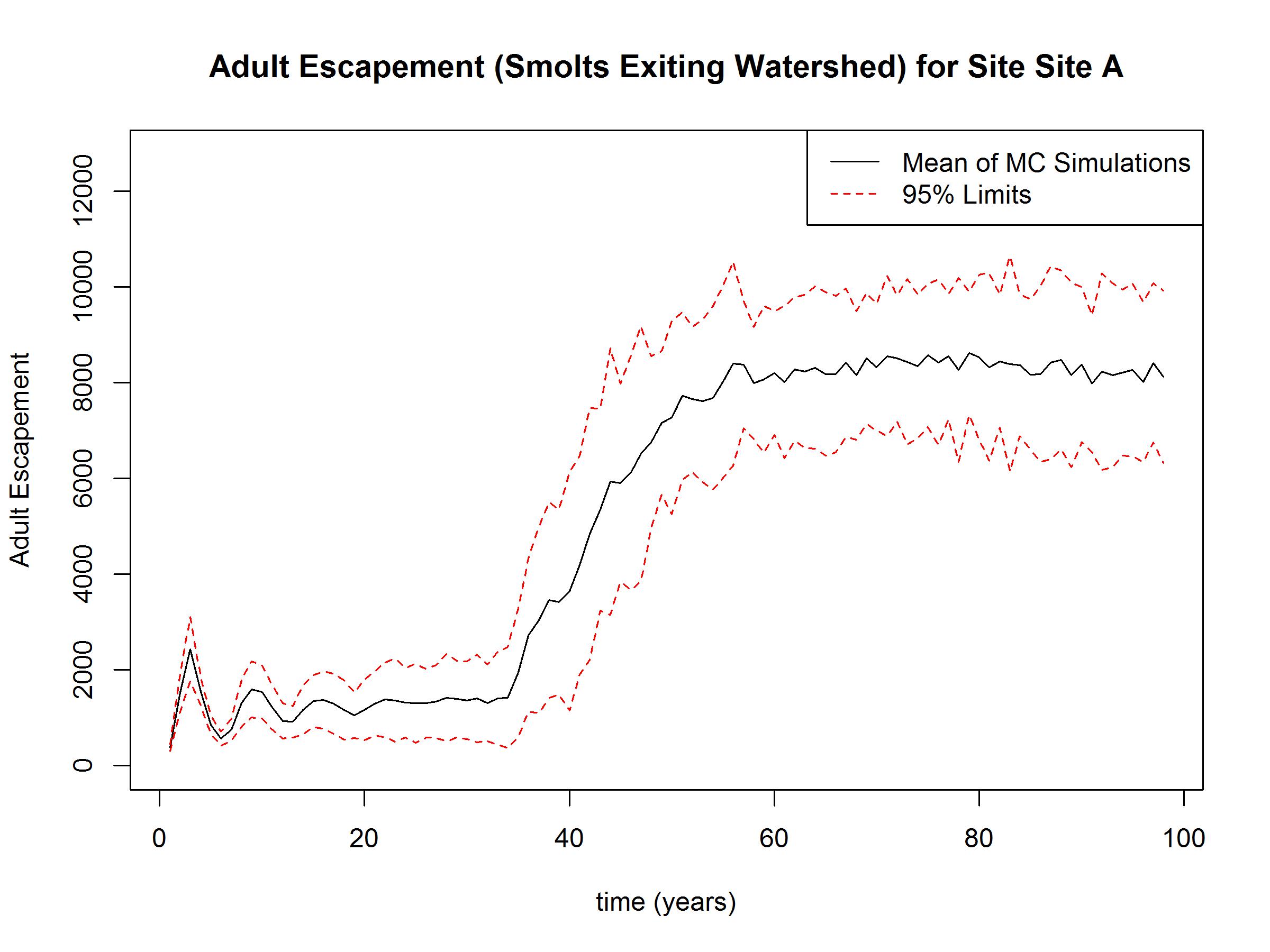
(Note: in actuality, some of these values are from the final time step, and some are from the 2nd to last time step. Due to the fact that some vales are calculated from the previous year’s time step, and some are calculated from the same year’s time step, the model cannot calculate all values through to the last simulation time step. This is accounted for, however, when using these files as input files to follow on simulations: Values are either assigned to “year-1” or “year-2”, as appropriate).

**10.2. Output Data Plots**

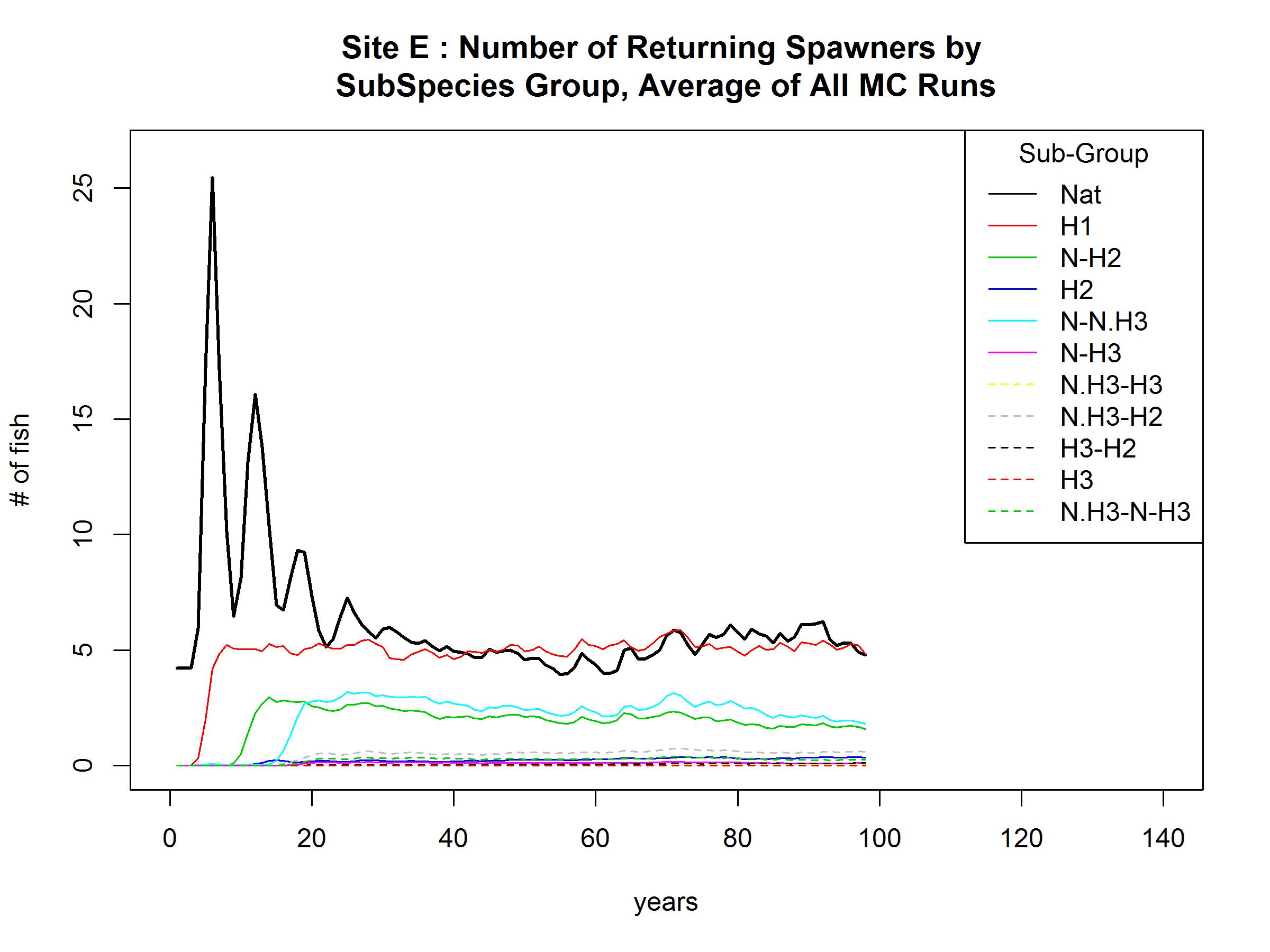
A series of output plots are generated. These are displayed on the screen, as well as stored as jpeg files in the folder “Output Plots”, which will be created as a subdirectory of the working directory. Standard plots created include the following:

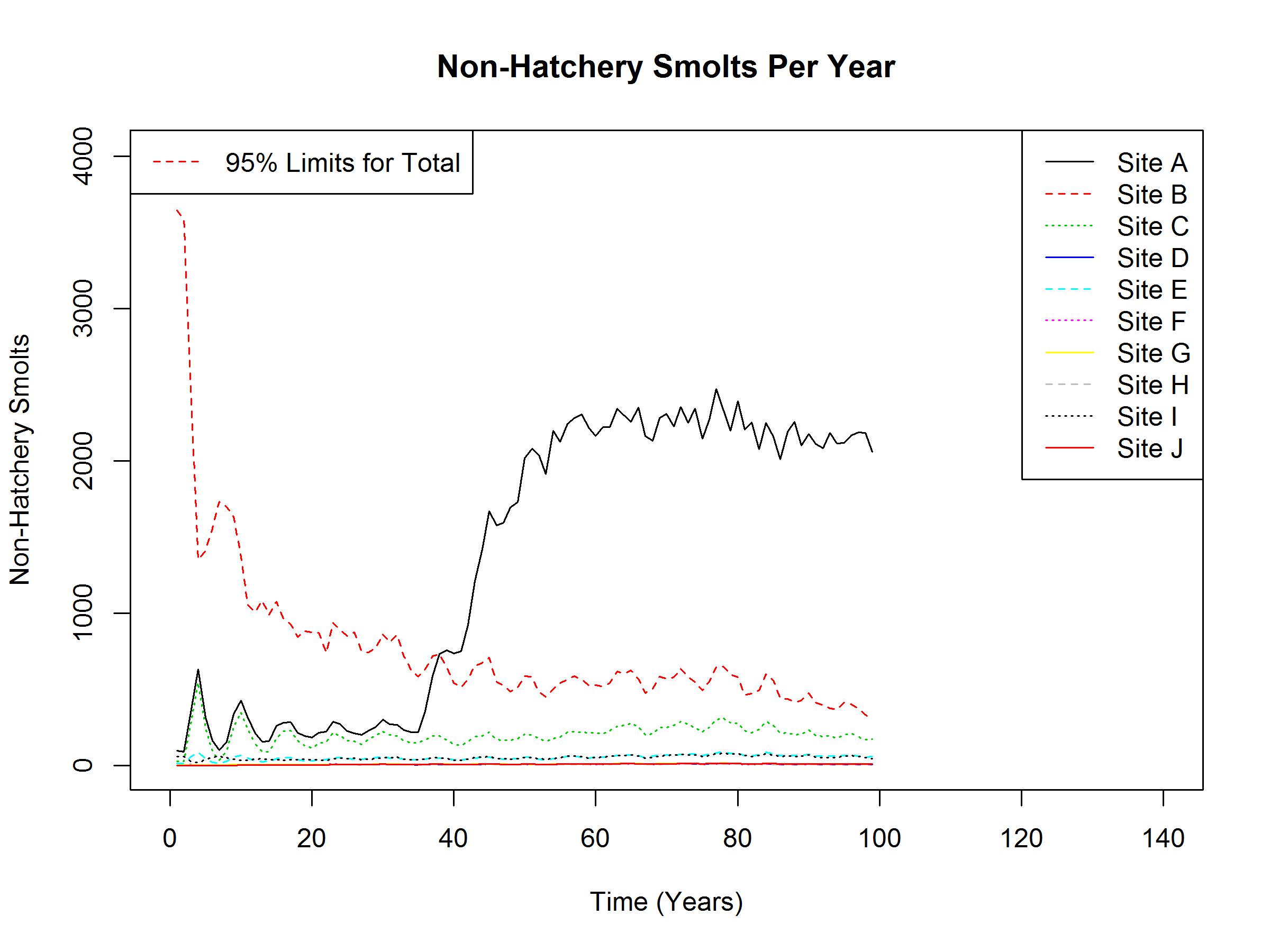
**Adult Escapement**: Figure 10.1 shows an example output plot for the number of smolts ***successfully*** migrating to the sea, across all genetic subtypes. The mean result, by time, across all Monte-Carlo iterations, is plotted. If Two or more Monte-Carlo iterations are modeled, 95% bounds on the distribution of results are included.

**Figure 10.1. Sample Model Output for Adult Escapement**



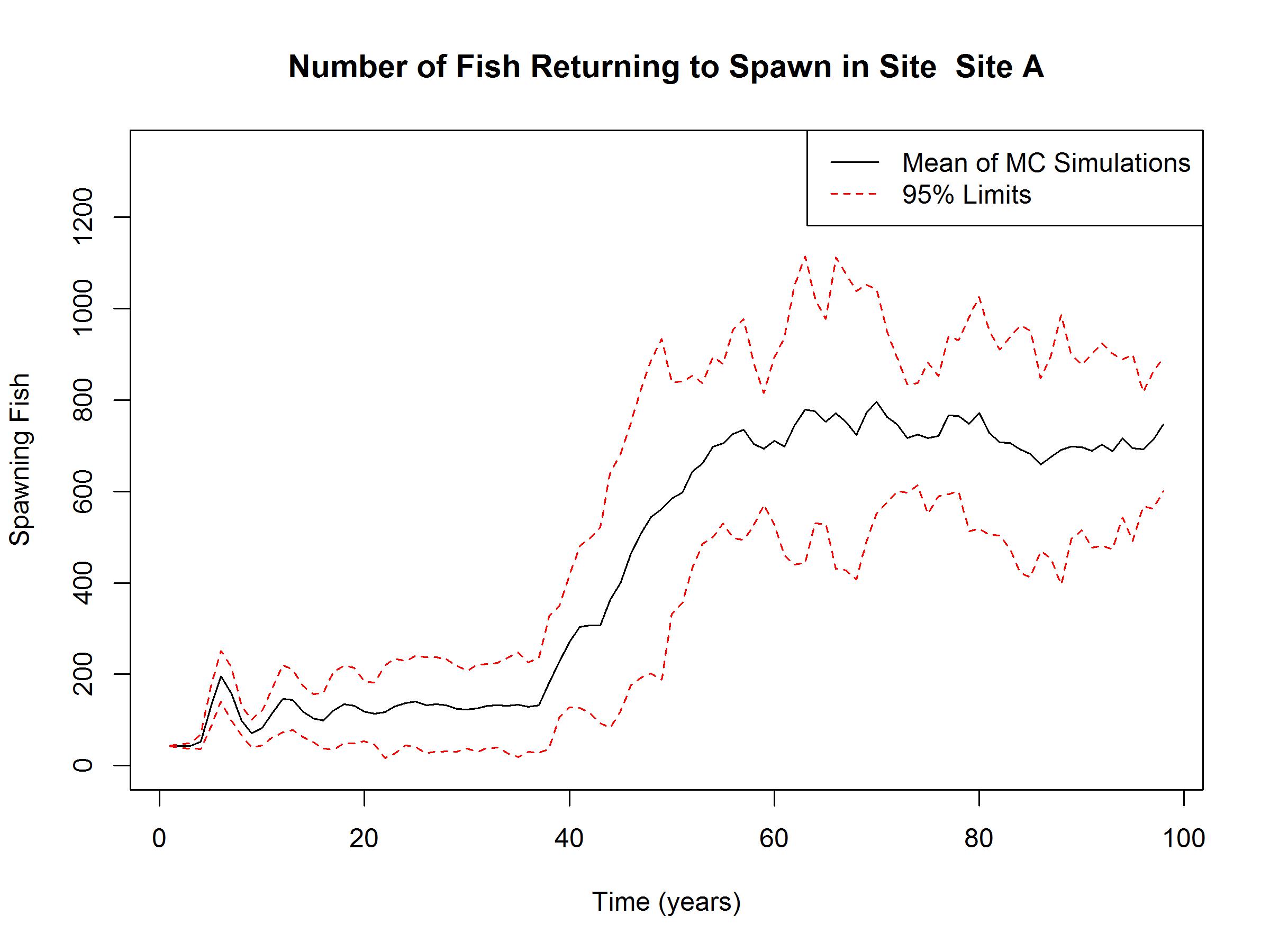
**Non-Hatchery Smolts:** Figure 10.2 shows an example output plot for non-hatchery smolts per year, by site. This includes all smolts except those of the “H1” genetic subtype (those individuals bred in a hatchery). Lines plotted represent the average results over all Monte-Carlo simulations. Each line represents an individual site.

**Figure 10.3. Returning Spawners by Site, by Genetic Sub-Type**

**Figure 10.2. Non Hatchery Smolts per Year**

**Returning Spawners by Site, by Genetic Sub-Type**: Figure 10.3 shows an example output plot of returning spawners for a site, by genetic sub-type.

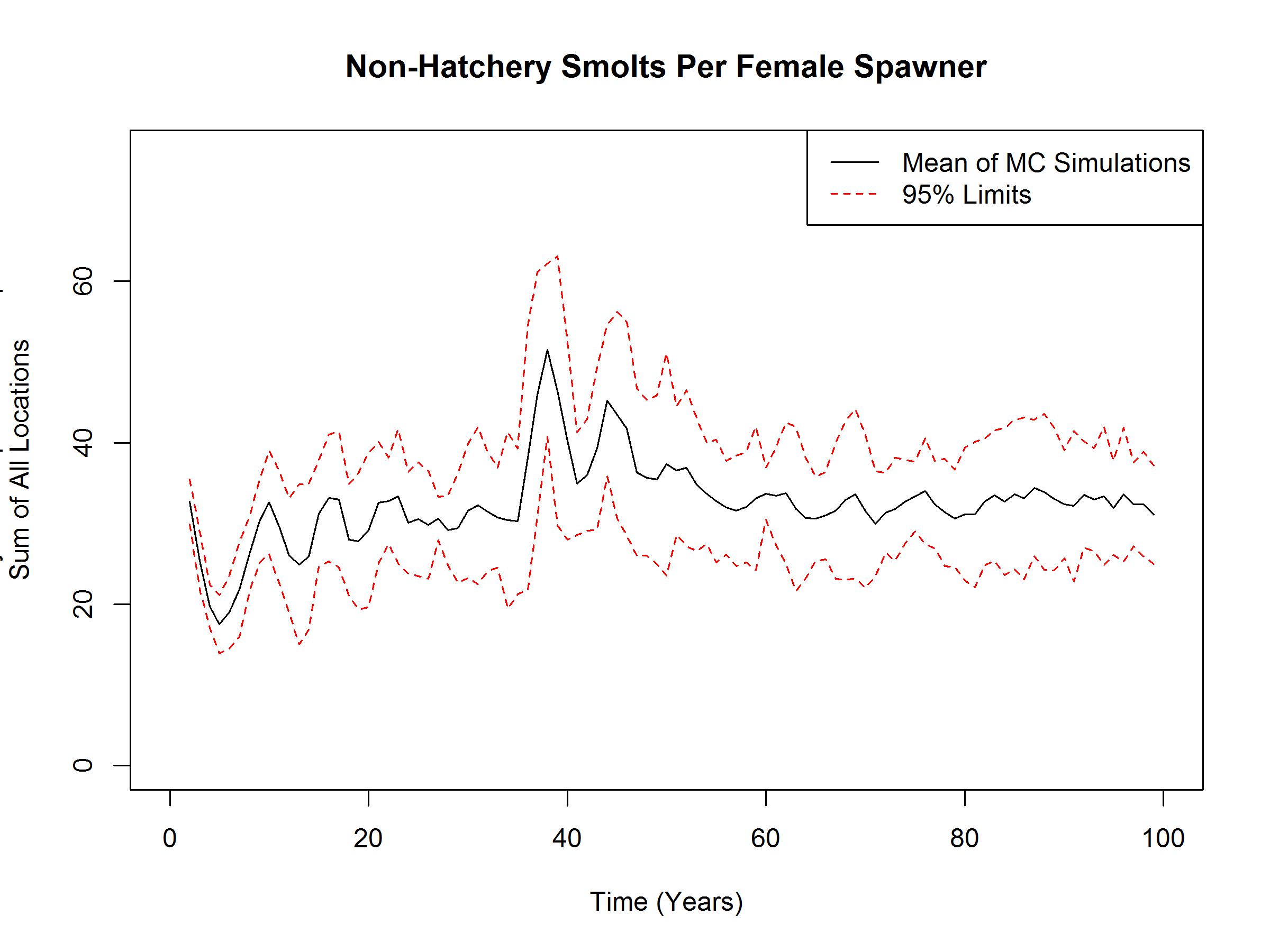
**Returning Spawners by Site, Total**: Figure 10.4 shows an example output plot of returning spawners for a site, totaled across all genetic sub-types, with limits showing the upper and lower 95% bounds on the distribution of results.

**Figure 10.4. Returning Spawners by Site, Sum of All Genetic Sub-Types**

**Smolts per Female Spawner**

Lastly, a plot of the average smolts per female spawner, totaled across the entire watershed, is included. Note that the smolts and spawners used in this calculation are not exact matches to spawners and their unique broods. Rather, all smolts (excluding those of hatchery origin) for a given year are divided by the number of female spawners in that same year to arrive at a smolts per spawner value. Thus, this response will be biased relative to year to year and near term trends, but should accurately reflect trends over the long term.

**Figure 10.5. Smolts per Female Spawner**



Users with competent R skills may also wish to access data and results for additional analysis, plotting, or create additional output files. Consult the author for details on data structures and related information necessary for custom plots.

1. **Checking the Inputs with Graphs**

An R file called “Watershed\_Graph\_Inputs.R” is included in package. This is not part of the watershed model, but rather a separate script available to allow users to automatically generate plots of all inputs versus simulation time. This may be especially useful to those users who are including stochasticity, step function changes, and/or trend changes in their parameter specifications.

This R-script uses functions in the files “Watershed\_Read\_Data.R” and “Watershed\_MonteCarlo.R” files. In addition, it uses all the input data files except the initial values files. Therefore, this file must be stored in the same folder as the other R files and the data input files. Also, the same add-on R packages required to run the watershed model are required to run this file (VGAM and MCMCpack). See section 2 of this document for details.

To run this script, simply open R, set the working directory to where the files are stored, and run the script by selecting all and running all (or use the shortcuts “ctrl-a” followed by “ctrl-r”. Depending on the number of sites and number of Monte-Carlo iterations, the code may take up to several minutes to run.

At its completion, it will generate a large number of graphs (with the total number dependent on the number of sites included in the model). The Page Up / Page Down keys can be used to scroll through the plots on the plot window. Plots are generated in roughly the same order as inputs are listed in the site level input files. The cross-site migration plots are last. If multiple Monte-Carlo iterations are included, a line for each Monte-Carlo iteration, for each metric included on a plot, is included.

Note that these are not intended as presentation quality plots, but rather as a useful tool to check that the inputs are as intended, and to visualize the variability and trend changes.

1. **Common Problems and Troubleshooting**

The watershed model is fairly complex, requiring a large number of user specified inputs. While efforts have been made to create a robust model and check that the model functions properly across all realistic ranges of data inputs and many corner cases, there will are undoubtedly some corner cases or user errors that will cause program crashes or produce erroneous results. In general, recommended troubleshooting is simply to close R and restart the program, and to check all values in all input files are present and are valid. The following list provides a debugging checklist covering common issues that may prevent the Watershed model from running successfully.

**Watershed Model: Debugging Checklist**

* Are all inputs entered in user file? Are there any blanks where number (even zeroes) should be?
* Are the input files saved as .csv file?
* Is the number of regions specified in "Watershed\_Header\_File.csv” consistent with the number of input sheets?
* Is the number of input data files per site specified in "Watershed\_Header\_File.csv” consistent with the number of input files?
* Is the number of habitat types specified in "Watershed\_Header\_File.csv” consistent with the number of habitat types in the input data files?
* Is the number of land use classifications specified in "Watershed\_Header\_File.csv” consistent with the number of habitat types in the input files?
* Are initial values specified for all life stages? Are files present for all sites? (Note: zero's are OK)
* Have cells been moved, or rows or columns added, to any input file? (Cells must remain exactly as in initial file templates)
* Are Excel and R files in the same directory, and is that directory set as the working directory within R?
* Has the program crashed or been stopped prior to completion? May need to close R and restart to ensure memory data structure integrity.
* Have all required R add-on packages been installed (see section 2).
* Are any output files from previous model runs being used by another program? (If so, the R-code will error as it tries to over-write a file in use).
* Do any of the site names include “/”? This will cause an error when attempting to wite output files that include the site names.

There are a set of warning messages that may appear when the program runs correctly. These have to do with the installation of packages that, in some cases, mask other functions loaded as default with R. These can be safely ignored. Table 12.1 lists these ignorable warning messages.

**Table 12.1. Warning Message that can be Ignored**



1. **References**
2. **Quantitative Consultants, Inc.** Salmon ISEMP watershed model development: Adding stochasticity to the Life History Model Structure. 2006
3. **M.D. Scheuerell, R. Hillborn, M.H. Ruckelshaus, K.K. Bartz, K.M.Lagueux, A.D. Hass, K. Rawson.** The Shirz Model: a Tool for Incorporating Anthropgenic Effects and Fish-Habitat Relationships in Conservation Planning. J. Fish Aquat. Sci. 63:1596-1607. 2006.
4. **Yuen, H. and Sharma, R**. 2005. Using simulation techniques to estimate management parameters on Snake River steelhead: Declines in productivity make rebuilding difficult. North Am. Jr. of Fish Mgt 25: 446-463.
5. **Beverton, R. J. H. and S. J. Holt. 1957**. *On the Dynamics of Exploited Fish Populations*. Chapman & Hall, New York.
6. **Mousalli, E. and R. Hilborn.** 1986. Optimal stock size and harvest rate in multi-stage life

history models. Canadian Journal of Fisheries and Aquatic Sciences 43:135-141.

1. **Pacific Northwest Hatchery Reform Project**, Columbia River Hatchery Reform Project, Final System-wide Report – Appendix C: Analytical Methods and Data Sources. <http://hatcheryreform.us/hrp_downloads/reports/columbia_river/system-wide/4_appendix_c_analytical_methods_and_info_sources.pdf>