

HATCHMARK2E

A statistical tool to estimate the proportion of hatchery-origin spawners using generalized least squares

Richard A. Hinrichsen and Rishi Sharma

June 28, 2012

Introduction

Assessments of the status of endangered Columbia River salmon populations, which include estimation of extinction probabilities and long-term trend of natural-origin fish, require reliable estimates of the proportion of hatchery-origin spawners on the spawning grounds (McClure et al. 2003). Furthermore, an assessment of the degree of interbreeding of hatchery-origin with the wild-origin segment, which may reduce the genetic fitness of subsequent generations, also depends on this estimate (Waples 1991). To allow distinction between natural-origin and hatchery-origin salmon in the Columbia Basin, the U.S. Congress presently requires the US Fish and Wildlife Service to visibly mark all hatchery production intended for harvest.¹ Visible marking of hatchery releases is a widespread practice among hatchery operators in the Columbia River basin, though non-visible marking procedures are sometimes substituted for or added to visible marks.

Despite the importance of estimates of proportion of hatchery-origin fish on the spawning grounds, reliable estimation techniques are lacking. The statistical difficulty of estimating the proportion of hatchery-origin escapement when hatchery-fish that are not visibly marked are present has been recognized for over thirty years (Hankin 1982). In Hankin's (1982) paper, it was assumed that all hatchery fish that were not coded-wire tagged were visibly marked at the same rate regardless of their hatchery of origin. This assumption certainly simplifies the analysis, but is not always realized in the Columbia Basin where source hatcheries are known to use different visible marking rates (Hinrichsen et al. 2012).

¹ On June 27, 2007, the House passed (amended) H.R. 2643, including a provision requiring the U.S. Fish and Wildlife Service to implement a system of mass marking of salmonid stocks that are released from federal hatcheries.

An estimator of the proportion of hatchery-origin fish is obtained for a program in which certain fractions of juveniles are visibly marked (VM), coded-wire tagged (CWT), or both VM and CWT. Some of the VM fish are then recovered as adults at a given spawning area along with fish that are the progeny of salmon spawning in the wild. The recoveries are fish that were sampled at the spawning grounds, usually in a carcass survey. Carcasses that have a VM are then checked for a coded wire tag and if one exists, the hatchery of origin is identified from the tag. In the case of a single hatchery input where VM fraction at that hatchery is known, it is a simple to derive an estimate of hatchery-origin spawners in the survey: the estimate is equal to the number of spawners that are not VM in the carcass survey divided the VM fraction. The estimate of the proportion of hatchery-origin spawners is then equal to this estimate of the number of hatchery-origin spawners in the survey divided by the number of carcasses surveyed. In the case of multiple hatchery inputs with different VM fractions, the estimator of proportion of hatchery-origin spawners is more complicated.

The goal of this work is to present a reliable estimator in the general case where there are spawners from multiple hatcheries and different source hatcheries may use different VM fractions. This general case becomes important whenever different VM fractions are applied for the hatcheries that supply inputs to spawning escapements in the wild. If a single VM fraction is applied to all of the hatcheries supplying inputs to spawning escapement, then this generalization is not needed. However, when different VM fractions are applied, it becomes necessary to estimate the number of spawners in the sample that come from each hatchery. This problem is solved using the method of moments which results in a generalized least squares (GLS) estimation problem (Kariya and Kurata 2004). Using theoretical formulas from the theory of GLS, we develop the GLS estimator (GLSE) of the proportion of hatchery-origin spawners and its variance. The formula for the variance may be written as a function of variables that may be controlled at the hatchery or in the spawning ground surveys: VM fractions, CWT fractions, and sampling rate. As these variables are increased, the precision and accuracy of the estimate of the proportion of hatchery-origin spawners also increase.

Statistical code for the analysis, written in the R programming language, may be found in Appendix A. For convenience, in the mathematical descriptions and derivations, we use an abbreviated set of variable names. The variable names used in the R code and the webtool HATCHMARK2E are given in Appendix B along with their definitions.

Methods

We generalize the estimation problem of Hankin (1982) to handle multiple hatcheries with potentially different VM fractions and different CWT fractions. This problem is more complicated than the problem of a single hatchery because the number of VM fish from a given hatchery must be estimated: it may no longer be treated as known because only a fraction of the VM fish are given a unique tag identifying the hatchery of origin. Let λ_i represent the VM fraction that is applied to hatchery fish released from hatchery i and let ϕ_i be the fraction of

these released fish that receive a CWT that uniquely identifies the hatchery of origin. Further assume that each returning fish has the same probability of being sampled, θ . The outline of the mathematical derivations and the equation is presented here. The assumptions for this estimation problem are given in Table 1.

Table 1. —Assumptions¹

(A1)	(Fixed probabilities) Hatchery-specific VM fractions and escapement sample rate are known.
(A2)	(Fixed probabilities) Hatchery-specific CWT fractions are known.
(A3)	(Identically distributed) Every individual spawner has the same probability of being sampled.
(A4)	(Identically distributed) Every individual hatchery-origin spawner from the same hatchery has the same probability of having a mark.
(A5)	(Independence) Whether any individual is sampled has no effect on the probability that another individual is sampled.
(A6)	(Independence) Whether any individual hatchery-origin spawners is observed to have a mark has no effect on the probability that another individual will have a mark.

¹ For convenience, we derived the estimators for releases grouped at the hatchery level. To split the data by release group instead, simply replace “hatchery” by “release” in the estimation method and interpret VM fractions and CWT fractions as release-specific. The GLSE we derived generalizes the estimator of Hankin (1982), who assumed two groups of releases: one that was VM and CWT at 100%, and another that used a constant VM fraction.

The assumptions in Table 1 allow us to express the joint distribution of escapement counts as a product of multinomial distributions:

$$f(\mathbf{x}) = \prod_{i=1}^n \binom{H_i}{x_{1,i}, x_{2,i}, x_{3,i}, x_{4,i}} (\theta \phi_i \lambda_i)^{x_{1,i}} (\theta \lambda_i (1 - \phi_i))^{x_{2,i}} (\theta (1 - \lambda_i))^{x_{3,i}} (1 - \theta)^{x_{4,i}} \quad (1)$$

$$\times \binom{W}{x_5} \theta^{x_5} (1 - \theta)^{W - x_5},$$

where H_i represents the hatchery-origin spawner escapement that originated in hatchery i , W represents the natural-origin spawner escapement, $x_{1,i}$ is the sampled and VM and CWT spawners from hatchery i , $x_{2,i}$ is the sampled and VM and not CWT spawners from hatchery i , $x_{3,i}$ is the number of sampled spawners from hatchery i that are not visibly marked, $x_{4,i}$ is the

unsampled spawners from hatchery i , x_5 is the sampled natural-origin spawners. Notice that $H_i = x_{i,1} + x_{i,2} + x_{i,3} + x_{i,4}$, which may be rearranged to give $x_{i,4} = H_i - x_{i,1} - x_{i,2} - x_{i,3}$.

Generalized least squares (GLS). —When there are multiple hatchery inputs, where not all VM fish are CWT, we apply the method of moments, which leads to an over-determined system of equations. This over-determined system is solved using GLS. To develop the system of equations, we use the method of moments, equating observed cell counts in the multinomial distributions to their expected values. Using this approach yields the following system of equations:

$$x_{1,i} = \theta\phi_i\lambda_i H_i, \quad (2)$$

and

$$x_2 = \sum_{i=1}^n \theta\lambda_i(1 - \phi_i)H_i, \quad (3)$$

where $x_2 = \sum_{i=1}^n x_{2,i}$ is the total observed spawners that are VM but do not have a CWT to identify their hatchery of origin. Considering equations (2) and (3) over all input hatcheries forms a system of $n+1$ equations with n unknowns (the hatchery-specific escapements): an over-determined system. One approach to solving this over-determined system uses GLS, which has a well-developed theory (Kariya and Kurata 2004). Because the observations are not all independent, we do not simply minimize the sum of squared differences between the observed observations and their expected values as in ordinary least squares. We instead minimize the square of the Mahalanobis distance given by

$$(\mathbf{x} - \mathbf{BH})' \Sigma^{-1} (\mathbf{x} - \mathbf{BH}), \quad (4)$$

where $\mathbf{x} = [x_{11} \ \dots \ x_{1n} \ x_2]'$ is the vector of observations, \mathbf{B} is a $(n+1) \times n$ matrix of cell count probabilities given by the partitioned matrix

$$\mathbf{B} = \begin{bmatrix} \mathbf{B}_1 \\ \mathbf{B}_2 \end{bmatrix}, \quad (5)$$

where $\mathbf{B}_1 = \text{diag}(\theta\lambda_1\phi_1, \dots, \theta\lambda_n\phi_n)$ is a diagonal $n \times n$ matrix, and $\mathbf{B}_2 = [\theta\lambda_1(1-\phi_1) \dots \theta\lambda_n(1-\phi_n)]$ is a row vector of dimension $1 \times n$. Σ is the covariance matrix for the vector of observations, \mathbf{x} , given by the partitioned matrix

$$\Sigma = \begin{bmatrix} \Sigma_{11} & \Sigma_{12} \\ \Sigma_{21} & \Sigma_{22} \end{bmatrix}, \quad (6)$$

where $\Sigma_{11} = \text{diag}(H_1\theta\lambda_1\phi_1(1-\theta\lambda_1\phi_1), \dots, H_n\theta\lambda_n\phi_n(1-\theta\lambda_n\phi_n))$ is a diagonal $n \times n$ matrix, $\Sigma_{12} = [H_1\theta^2\lambda_1^2\phi_1(1-\phi_1) \dots H_n\theta^2\lambda_n^2\phi_n(1-\phi_n)]'$ is an $n \times 1$ vector, $\Sigma_{21} = \Sigma'_{12}$, and $\Sigma_{22} = \sum_{i=1}^n H_i\theta\lambda_i(1-\phi_i)(1-\theta\lambda_i(1-\phi_i))$ is a scalar. With these definitions, it is then a matter of using the well-known solution to the minimization problem (treating the variance as fixed)

$$\hat{\mathbf{H}} = (\mathbf{B}'\Sigma^{-1}\mathbf{B})^{-1}\mathbf{B}'\Sigma^{-1}\mathbf{x}. \quad (7)$$

The variance matrix for the GLSEs is given by

$$\text{var}(\hat{\mathbf{H}}) = (\mathbf{B}'\Sigma^{-1}\mathbf{B})^{-1}. \quad (8)$$

Using the above definitions, it may be shown that the GLSE for the number of hatchery-origin spawners from hatchery i is given by

$$\hat{H}_i = \frac{x_{1,i}}{\theta\lambda_i\phi_i} + \frac{\frac{H_i(1-\phi_i)}{\phi_i}}{\sum_{i=1}^n \left(\frac{H_i(1-\phi_i)\theta\lambda_i}{\phi_i} \right)} \left(x_2 - \sum_{i=1}^n \frac{x_{1,i}(1-\phi_i)}{\phi_i} \right), \quad (9)$$

where it is assumed that not all of the CWT fractions are 1, none of the CWT fractions is zero, none of VM fractions is zero, and sampling rate is not zero; otherwise, equation (9) is undefined. The special case of 100% CWT of all VM fish is handled later in equations (21)-(25). Continuing with the method of moments, the estimate of the total spawning escapement is

$$\hat{E} = (E_U + E_M) / \theta, \quad (10)$$

where E_U represents the observed spawners that are not VM, and E_M represents the observed VM spawners.

When all hatcheries use the same VM fraction, equation (9) reduces to

$$\hat{H} = \sum_{i=1}^n \hat{H}_i = \frac{E_M}{\theta\lambda}, \quad (\text{assuming } \lambda_i = \lambda \text{ for } i = 1, \dots, n) \quad (11)$$

where E_M is the observed number of VM spawners (with or without CWTs).

The proportion of hatchery-origin spawners may be estimated using the GLSEs of hatchery-origin escapement and total escapement as follows:

$$\hat{p} = \frac{\hat{H}}{\hat{E}}, \quad (12)$$

where $\hat{E} = \hat{H} + \hat{W}$ is the estimate of the total spawning population.

Using equation (8) along with the fact that $H = \sum_{i=1}^n H_i$, we may write

$$\text{var}(\hat{H}) = \sum_{i=1}^n \frac{H_i(1 - \theta\lambda_i\phi_i)}{\theta\lambda_i\phi_i} - \frac{\left(\sum_{i=1}^n \frac{H_i(1 - \phi_i)}{\phi_i} \right)^2}{\sum_{i=1}^n \frac{H_i(1 - \phi_i)\theta\lambda_i}{\phi_i}}. \quad (13)$$

It was assumed in the above formula for $\text{var}(\hat{H})$ that not all of the CWT fractions were equal to 1. In that event, the formula is undefined. The case of 100% CWT of all VM fish is handled later in equations (21)-(25).

In the special case of a constant marking rate, the variance of the hatchery-origin escapement becomes

$$\text{var}(\hat{H}) = \sum_{i=1}^n \frac{H_i(1 - \theta\lambda)}{\theta\lambda}, \quad (\text{assuming } \lambda_i = \lambda \text{ for } i = 1, \dots, n) \quad (14)$$

which involves no CWT fractions and is equivalent to the single-hatchery case.

We now return to the general case. The main focus on this work is not the variance of \hat{H} , but the variance of \hat{p} . For this we use the technique of a Taylor Series expansion of p about the GLSEs \hat{H} and \hat{E} . It will then become apparent that the variance of \hat{p} may be written as a function of the true value of p , E , the variances of \hat{H} and \hat{E} and their covariance. Using the multinomial distributions defined in equation (1), we may write

$$\begin{aligned} \text{var}(\hat{E}) &= \text{var}\left(\sum_{i=1}^n \left(\frac{H_i - x_{4,i}}{\theta}\right) + \frac{x_5}{\theta}\right) = \sum_{i=1}^n \left(\frac{\text{var}(x_{4,i})}{\theta^2}\right) + \frac{\text{var}(x_5)}{\theta^2} \\ &= \frac{H(1-\theta)\theta + W(1-\theta)\theta}{\theta^2} \\ &= \frac{E(1-\theta)}{\theta}. \end{aligned} \tag{15}$$

$$\begin{aligned} \text{cov}(\hat{H}, \hat{E}) &= \text{cov}\left(\sum_{i=1}^n \frac{x_{1,i}}{\theta \lambda_i \phi_i} + \frac{m}{\theta} \left(x_2 - \sum_{i=1}^n \frac{x_{1,i}(1-\phi_i)}{\phi_i}\right), \sum_{i=1}^n \left(\frac{H_i - x_{4,i}}{\theta}\right) + \frac{x_5}{\theta}\right) \\ &= \frac{-1}{\theta^2} \sum_{i=1}^n \left(\frac{1}{\lambda_i \phi_i} - m \frac{(1-\phi_i)}{\phi_i}\right) \text{cov}(x_{1,i}, x_{4,i}) + m \text{cov}(x_2, x_{4,i}) \\ &= \frac{(1-\theta)}{\theta} H, \end{aligned} \tag{16}$$

where

$$m = \frac{\sum_{i=1}^n \frac{H_i(1-\phi_i)}{\phi_i}}{\sum_{i=1}^n \frac{H_i(1-\phi_i)\lambda_i}{\phi_i}}. \quad (17)$$

With these variance and covariance formulas it is now possible to derive the variance of the estimate of the proportion of hatchery-origin spawners. Using a first-order Taylor series expansion, we may write

$$\hat{p} - p \cong (\nabla p)' \begin{bmatrix} \hat{H} - H \\ \hat{E} - E \end{bmatrix}, \quad (18)$$

$$\text{var}(\hat{p}) \cong (\nabla p)' \text{var} \begin{bmatrix} \hat{H} \\ \hat{E} \end{bmatrix} \nabla p \quad (19)$$

$$= \frac{1}{E} \left\{ \sum_{i=1}^n \frac{p_i(1-\theta\lambda_i\phi_i)}{\theta\lambda_i\phi_i} - \frac{\left(\sum_{i=1}^n \frac{p_i(1-\phi_i)}{\phi_i} \right)^2}{\sum_{i=1}^n \frac{p_i(1-\phi_i)\theta\lambda_i}{\phi_i}} - p^2 \frac{(1-\theta)}{\theta} \right\},$$

where $p_i = H_i / E$. It was assumed in the above formula for $\text{var}(\hat{p})$ that not all of the CWT fractions were equal to 1. In that event, the variance formula in equation (19) is undefined. The special case of CWT of all VM fish is treated in equations (21)-(25). The theoretical variance formula is useful because it shows clearly how the variance is related to the sampling rate, VM fractions, and CWT fractions.

Notice that in the special case where the VM fraction is constant, equation (19) reduces to

$$\text{var}(\hat{p}) = \frac{p}{E} \left\{ \frac{(1-\lambda\theta)}{\lambda\theta} - p \frac{(1-\theta)}{\theta} \right\}. \quad (\text{assuming } \lambda_i = \lambda \text{ for } i = 1, \dots, n) \quad (20)$$

Special case (all VM fish given a CWT). —In the special case where all VM hatchery fish are given a coded wire tag, the hatchery-specific escapements are given by

$$\hat{H}_i = \frac{x_{1,i}}{\theta\lambda_i}. \quad (\text{assuming } \phi_i = 1 \text{ for } i = 1, \dots, n) \quad (21)$$

Using equation (1) it is easily shown that

$$\text{var}(\hat{H}_i) = \frac{\text{var}(x_{1,i})}{(\theta\lambda_i)^2} = \frac{H_i(1-\theta\lambda_i)}{\theta\lambda_i}, \quad (\text{assuming } \phi_i = 1 \text{ for } i = 1, \dots, n) \quad (22)$$

$$\text{var}(\hat{E}) = \text{var}\left(\sum_{i=1}^n \frac{H_i - x_{4,i}}{\theta} + \frac{x_5}{\theta}\right) = E \frac{(1-\theta)}{\theta}, \quad (\text{assuming } \phi_i = 1 \text{ for } i = 1, \dots, n) \quad (23)$$

$$\begin{aligned} \text{cov}(\hat{H}, \hat{E}) &= \text{cov}\left(\sum_{i=1}^n \frac{x_{1,i}}{\theta\lambda_i}, \sum_{i=1}^n \frac{H_i - x_{4,i}}{\theta} + \frac{x_5}{\theta}\right) \\ &= H \left(\frac{1-\theta}{\theta} \right), \end{aligned} \quad (\text{assuming } \phi_i = 1 \text{ for } i = 1, \dots, n) \quad (24)$$

and

$$\text{var}(\hat{p}) \equiv (\nabla p)' \text{var} \begin{bmatrix} \hat{H} \\ \hat{E} \end{bmatrix} \nabla p \quad (\text{assuming } \phi_i = 1 \text{ for } i = 1, \dots, n) \quad (25)$$

$$= \frac{1}{E} \left\{ \sum_{i=1}^n \frac{p_i(1 - \theta \lambda_i)}{\theta \lambda_i} - p^2 \frac{(1 - \theta)}{\theta} \right\}.$$

In all cases, the variance of the wild-origin escapement estimate is calculated using the equation

$$\text{var}(\hat{W}) = \text{var}(\hat{E} - \hat{H}) = \text{var}(\hat{E}) + \text{var}(\hat{H}) - 2\text{cov}(\hat{E}, \hat{H}). \quad (26)$$

Theoretical estimates of CV and SE in HATCHMARK2E were calculated using the theoretical variance formulas derived above. Alternatively, CV and SE are estimated using Bootstrapping.

Bootstrapping. — To evaluate the relative bias and precision of these GLSEs, bootstrapping was used. Using bootstrapping as an alternative to the theoretical estimates of precision can be important when sample size is low and the asymptotic properties may not apply. The idea is to assume some true values of the hatchery-origin and natural-origin escapements, simulate the data collection process again and again and use these simulated data and true values to gauge the precision and accuracy of the escapement estimates. The underlying assumptions are that a spawning fish is sampled with probability θ , the probability that an observed hatchery fish from hatchery i is VM is λ_i , and the probability that a VM fish from hatchery i is also CWT is ϕ_i . Using binomial random variables, we generate M bootstrap replications of the estimate the proportion of spawners that are of hatchery origin. The bootstrap replications are then used to determine that statistical properties of the GLSE, including: standard error, coefficients of variation, and relative bias. Bias is calculated as a relative bias, which is the true bias divided by the true value of the parameter estimated. Besides the special cases where (a) all the VM fish are given a coded wire tag or (b) all hatcheries mark the same fraction of releases, there two more special cases to consider in the estimation of \hat{p} . Whenever $x_{1,i} = 0$; $i = 1, \dots, n$ and $x_2 = 0$, we use the estimate $\hat{p} = 0$. Assuming special cases (a) and (b) do not hold and $x_{1,i} = 0$; $i = 1, \dots, n$

and $x_2 \neq 0$, \hat{p} is assumed to be unestimable because there is no information in the data set that may be used to divide the x_2 returns by hatchery of origin.

Given the 10,000 bootstrap replications of the estimate of the hatchery proportion of spawning escapement, denoted by $\hat{p}_1^*, \hat{p}_2^*, \dots, \hat{p}_M^*$, where $M = 10,000$ represents the number of bootstrap replications², the bootstrap estimate of variance is given by

$$\text{var}^*(\hat{p}) = \sum_{j=1}^M \frac{(\hat{p}_j^* - \bar{\hat{p}}^*)^2}{M-1}, \quad (27)$$

where $\bar{\hat{p}}^*$ represents the sample mean of the M bootstrap estimates. Bias is calculated as relative bias, namely,

$$\text{bias}^*(\hat{p}) = \frac{(\bar{\hat{p}}^* - \hat{p})}{\hat{p}}, \quad (28)$$

where

$$SE^*(\hat{p}) = \sqrt{\text{var}^*(\hat{p})}. \quad (29)$$

The coefficient of variation is then estimated as

² The number of bootstrap replications may be set by the user of HATCHMARK2E.

$$CV^*(\hat{p}) = \frac{SE^*(\hat{p})}{p}. \quad (30)$$

Hanford Reach application

To demonstrate the use of the GLSE, we applied it to the 2010 Hanford Reach fall Chinook salmon carcass survey data. The Hanford Reach is a 90-km stretch of the Columbia River extending from the upper end of Lake Wallula (created by McNary Dam) to Priest Rapids Dam (river km 639). Annual carcass surveys are used to estimate the proportion of hatchery-origin spawners in this reach. Carcasses are collected primarily by boat but foot surveys along the shorelines are also conducted. Carcasses with a visible mark (adipose fin clip) are tested with a hand-held CWT detection wand to determine whether a CWT is present. Carcasses without a visible mark are considered part of the total numbers sampled but are not checked for CWTs. (Note that this protocol may be improved by checking all sampled spawners for a visible mark [see Improved Protocol].) If a carcass contains a CWT, its snout (containing the CWT) is removed, frozen, and sent to the Washington Department of Fish and Wildlife coded-wire tag lab in Olympia, WA, to be examined. At the lab, the CWT is extracted and the CWT code is read to identify hatchery of origin. CWT codes are traced to the hatchery and release group of origin by biologists. This information is accessible via the Pacific States Marine Fisheries Commission's Regional Mark Processing Center at www.rmhc.org (August 2011).

In 2010, the hatcheries contributing spawners with CWTs in the Hanford Reach were Little White Salmon brood year (BY) 2005, Lyons Ferry (Snake River) BY 2006, Priest Rapids BY 2005 and BY 2007, Ringold Springs BY 2006 and BY 2007, and Umatilla BY 2007 (Table 2). For convenience, each hatchery-BY pair was treated as a separate hatchery in the equations describing the estimators and their variances. Thus there were a total of seven input hatchery-BY pairs ($n=7$). Of the 9,791 carcasses sampled in 2010, 23 had a visible mark and a CWT inserted at a hatchery, and 308 had a visible mark, but no CWT. The estimated sample rate, θ , was 0.11252. Some wild-born Hanford Reach juveniles were also VM'd and CWT'd and recovered as adults in the 2010 carcass survey. These wild-born recoveries (19 total) were not counted as hatchery-origin spawners in our estimation; instead, they were treated as a part of the unmarked sample.

Standard errors for GLSEs H and p were calculated as the square root of the theoretical variances where the unknown parameters in these formulas were replaced with their estimates. Relative biases were estimated using bootstrapping with 100,000 bootstrap replications of the estimates of the proportion of hatchery-origin spawners.

Results. — The GLSE of p for the Hanford Reach was 0.0766 (SE= 0.0090) The GLSE of the proportion of hatchery-origin spawners had a CV of 0.118 and an absolute relative bias < 0.3%. The GLSE of total hatchery fish was 6,668.1 (SE=788.9).

Table 2.—Visible marking and coded-wire tagging at source hatcheries that provide spawner inputs to Hanford Reach spawning grounds. The total number of spawning ground carcasses sampled in 2010 was 9,791 and the sample rate was 0.11252. Of the carcasses sampled, 23 were VM and CWT at a hatchery and 308 were VM only. The total number released may be calculated by summing the columns “VM & CWT,” “VM only,” “CWT only,” and “Not VM & not CWT.”

#	Hatchery	Brood year	VM & CWT	VM only	CWT only	Not VM & not CWT	VM fraction, λ	CWT fraction, ϕ	No. tags in sample
1	Little White Salmon NFH	2005	448,145	1,354,029	0	0	1.00	0.25	1
2	Priest Rapids H	2005	199,445	1,628,614	0	5,048,231	0.27	0.11	3
		2007	202,568	813	0	4,344,925	0.04	1.00	7
3	Ringold Springs H	2006	222,706	0	0	3,179,824	0.07	1.00	2
		2007	221,951	2,230,190	0	645,308	0.79	0.09	7
4	Lyons Ferry H ^a	2006	231,534	1,673	220,350	6,076	0.51	0.99	1
5	Umatilla H	2007	279,480	0	0	0	1.00	1.00	2

^aSome Lyons Ferry Hatchery juveniles were visually marked with a visual implant elastomer (VIE) tag. In this application, however, only fish that were marked with an adipose fin clip were considered visually marked.

Acknowledgements

The authors gratefully acknowledge the reviews of Amber Parsons, Charlie Paulsen, Tracy Hillman, Bruce Crawford, Michael Newsome, Craig Busack, Brian Maschhoff, and Tim Fisher. Paul Hoffarth supplied the data from the Hanford Reach application. This work was supported by Bonneville Power Administration. The views expressed are solely those of the authors and are not intended to represent the views of any organization with which the authors are affiliated.

References

- Hankin, D.G. 1982. Estimating escapement of pacific salmon: marking practices to discriminate wild and hatchery fish. Transactions of the American Fisheries Society 111:286-298.
- Hinrichsen, R.A., R. Sharma, and T.R. Fisher. 2012. Precision and accuracy of estimators of the proportion of hatchery-origin spawners. Transactions of the American Fisheries Society 141:437-454.
- Kariya, T. and H. Kurata. 2004. Generalized Least Squares. Wiley Series in Probability and Statistics. Wiley. New York, New York.
- McClure, M.M., E.E. Holmes, B.L. Sanderson, and C.E. Jordan. 2003. A large-scale multispecial status assessment: anadromous salmonids in the Columbia River basin. Ecological Applications 13:964-989.
- Waples, R.S. 1991. Genetic interactions between hatchery and wild salmonids: lessons from the Pacific Northwest. Canadian Journal of Fisheries and Aquatic Sciences. 48(Suppl. 1): 124-133.

Appendix A R-code used to estimate the proportion of hatchery-origin spawners

```
#HATCHMARK2E Program to estimate proportion of hatchery-origin escapement using
#generalized least squares (Hinrichsen et al. 2011). Precision results are obtained using
#Bootstrapping or theoretical results
#
#This code treats the general case of inputs from several source hatcheries with potentially
#different visual marking fractions (VM fractions).

#Variables and parameters used in the analysis
#inputs
#Nsim = total number of bootstrap replications
#x2 = visibly marked and not coded-wire tagged observation
#x1 = visibly marked and coded-wire tagged observations (hatchery-specific)
#Eu = Number of spawners in the sample that are not visibly marked
#theta = sampling fraction
#lambda = marking rate (lambda) (hatchery-specific)
#phi=fraction of marked fish that are also coded-wire tagged (hatchery-specific)
#
#output variables
#Nnos = GLSE estimate of natural origin spawning escapement
#Nhos = GLSE estimate of hatchery origin spawning escapement (hatchery-specific)
#phos = GLSE estimate of the proportion of hatchery-origin spawning escapement
#SE.Nnoshat = standard error (SE) of Nnoshat
#CV.Nnoshat = Coefficient of variation of Nnoshat
#SE.Nhoshat = standard error (SE) of Nhoshat
#CV.Nhoshat = Coefficient of variation of Nhoshat
#SE.phoshat = standard error (SE) of the phos estimator
#CV.phoshat = Coefficient of variation of the phos estimator
#BIAS.phoshat = relative bias of the phos estimator

#the following use theoretical formulas

#SE2.Nnoshat = standard error (SE) of Nnoshat
#CV2.Nnoshat = Coefficient of variation of Nnoshat
#SE2.Nhoshat = standard error (SE) of Nhoshat
#CV2.Nhoshat = Coefficient of variation of Nhoshat
#SE2.phoshat = standard error (SE)
#CV2.phoshat = Coefficient of variation

#use bootstrapping for variance and bias
phos.mhatch.main1<-function(Nsim=10000,x1=c(40,23),x2=40,Eu=200,theta=0.25,
lambda=c(0.75,0.25),phi=c(.5,.9)){
```

```

#check inputs
k1<-length(x1);k2<-length(lambda);k3<-length(phi)
mytest<-abs(k1-k2)+abs(k2-k3)
if(mytest>0) stop("dimensions of x1, lambda, and phi must match")
nhatch<-length(x1)

if(theta>1)stop("theta must be less than or equal to one")
if(theta<=0)stop("theta must be greater than zero")
if(sum(lambda>1))stop("lambdas must all be less than one")
if(sum(lambda<=0))stop("lambdas must all be greater than zero")
if(sum(phi>1))stop("phis must all be less than one")
if(sum(phi<=0))stop("phis must all be greater than zero")

#check lambdas (if they are all the same, the analysis simplifies)
lambdatest<-FALSE
if(nhatch==1){lambdatest==TRUE}
if(nhatch>1){lambdatest<-var(lambda)<1.e-10}

if(lambdatest){
  Nhos<-(sum(x1)+x2)/(theta*lambda[1])
  Nnos<-(sum(x1)+x2+Eu)/theta - Nhos
  res1<-phos.estimates1(Nsims=Nsims, Nnos=Nnos,Nhos=Nhos,theta=theta,lambda=lambda[1])
}else{
  if((sum(abs(x1))<1.e-10)&(x2>0))stop("Nhos unestimable because lambdas differ and x1=0 and x2>0")
  phitest<-FALSE
  if(sum(phi==1)==nhatch)phitest<-TRUE
  if(!phitest){
    Nhos<-get.nhosat.all(x1=x1,x2=x2,theta=theta,lambda=lambda,phi=phi)
  }else{
    if(x2>0)stop("Error: All phis are one and x2 is greater than zero")
    Nhos<-x1/(theta*lambda)
  }
  Nnos<-(sum(x1)+x2+Eu)/theta - sum(Nhos)

  res1<-phos.mhatch.estimates1(Nsims=Nsims,Nnos=Nnos,Nhos=Nhos,theta=theta,
    lambda=lambda,phi=phi)
}

return(list(Nsims=res1$Nsims,
  x1=x1,
  x2=x2,
  Eu=Eu,
  theta=theta,

```

```

    lambda=lambda,
    phi=phi,
    Nnos=res1$Nnos,
    Nhos=sum(res1$Nhos),
    phos=res1$phos,
    SE.Nnoshat=res1$SE.Nnoshat,
    CV.Nnoshat=res1$CV.Nnoshat,
    SE.Nhoshat=res1$SE.Nhoshat,
    CV.Nhoshat=res1$CV.Nhoshat,
    SE.phoshat=res1$SE.phoshat,
    CV.phoshat=res1$CV.phoshat,
    BIAS.phoshat=res1$BIAS.phoshat))
}

#use theoretical variances formulas
phos.mhatch.main2<-function(x1=c(40,23),x2=40,Eu=200,theta=0.25,
lambda=c(0.75,0.25),phi=c(.5,.9)){

#check inputs
k1<-length(x1);k2<-length(lambda);k3<-length(phi)
mytest<-abs(k1-k2)+abs(k2-k3)
if(mytest>0) stop("dimensions of x1, lambda, and phi must match")
nhatch<-length(x1)

if(theta>1)stop("theta must be less than or equal to one")
if(theta<=0)stop("theta must be greater than zero")
if(sum(lambda>1))stop("lambdas must all be less than one")
if(sum(lambda<=0))stop("lambdas must all be greater than zero")
if(sum(phi>1))stop("phis must all be less than one")
if(sum(phi<=0))stop("phis must all be greater than zero")

#check lambdas (if they are all the same, the analysis simplifies)
lambdatest<-FALSE
if(nhatch==1){lambdatest==TRUE}
if(nhatch>1){lambdatest<-var(lambda)<1.e-10}

if(lambdatest){
  Nhos<-(sum(x1)+x2)/(theta*lambda[1])
  Nnos<-(sum(x1)+x2+Eu)/theta - Nhos
  res2<-phos.estimates2(Nnos=Nnos,Nhos=Nhos,theta=theta,lambda=lambda[1])
}else{
  if((sum(abs(x1))<1.e-10)&(x2>0))stop("Nhos unestimable because lambdas differ and x1=0 and
x2>0")

```

```

phitest<-FALSE
if(sum(phi==1)==nhatch)phitest<-TRUE
if(!phitest){
Nhos<-get.nhoshat.all(x1=x1,x2=x2,theta=theta,lambda=lambda,phi=phi)
}else{
if(x2>0)stop("Error: All phis are one and x2 is greater than zero")
Nhos<-x1/(theta*lambda)
}

Nnos<-(sum(x1)+x2+Eu)/theta - sum(Nhos)
res2<-phos.mhatch.estimates2(Nnos=Nnos,Nhos=Nhos,theta=theta,
lambda=lambda,phi=phi)
}

return(list(Nsims=NA,
x1=x1,
x2=x2,
Eu=Eu,
theta=theta,
lambda=lambda,
phi=phi,
Nnos=res2$Nnos,
Nhos=sum(res2$Nhos),
phos=res2$phos,
SE2.Nnoshat=res2$SE2.Nnoshat,
CV2.Nnoshat=res2$CV2.Nnoshat,
SE2.Nhoshat=res2$SE2.Nhoshat,
CV2.Nhoshat=res2$CV2.Nhoshat,
SE2.phoshat=res2$SE2.phoshat,
CV2.phoshat=res2$CV2.phoshat,
BIAS2.phoshat=NA))
}

#uses Bootstrapping for multiple hatcheries
#uses cwt ratios to help estimate fractions of
#unmarked fish from hatchery i

#Use Bootstrapping for results
phos.mhatch.estimates1<-function(Nsims=10000,Nnos=200,Nhos=c(100,100),theta=0.25,
lambda=c(0.75,.25),phi=c(.5,.9)){

#check dimension of inputs
k1<-length(Nhos);k2<-length(lambda);k3<-length(phi)
mytest<-abs(k1-k2)+abs(k2-k3)

```

```

if(mytest>0) stop("dimensions of Nhos, lambda, and phi must match")
nhatch<-length(Nhos)

#check inputs
if(sum(Nhos<0))stop("An Nhos estimate is negative")
if(Nnos<0)stop("Nnos estimate is negative")

#check lambdas (if they are all the same, the analysis simplifies)
mytest<-FALSE
if(nhatch==1){mytest==TRUE}
if(nhatch>1){mytest<-var(lambda)<1.e-10}
if(mytest){
#phis don't matter at all – it's as if there were a single hatchery
res<-phos.estimates1(Nsims,Nnos=Nnos,Nhos=sum(Nhos),theta=theta,lambda=mean(lambda))
phos=sum(Nhos)/(sum(Nhos)+Nnos)
myres<-list(Nsims=Nsims,
            Nnos=Nnos,
            Nhos=Nhos,
            theta=theta,
            lambda=lambda,
            phi=phi,
            phos=phos,
            SE.Nnoshat=res$SE.Nnoshat,
            CV.Nnoshat=res$CV.Nnoshat,
            SE.Nhoshat=res$SE.Nhoshat,
            CV.Nhoshat=res$CV.Nhoshat,
            SE.phoshat=res$SE.phoshat,
            CV.phoshat=res$CV.phoshat,
            BIAS.phoshat=res$BIAS.phoshat)

return(myres)
}

#check phis (must all exceed zero)
if(sum(phi==0))stop("phis must all be greater than zero")
phitest<-FALSE
if(sum(phi==1)==nhatch)phitest<-TRUE

phos<-sum(Nhos)/(sum(Nhos)+Nnos)
#generate synthetic data sets
Ehatchsampled<-matrix(NA,nrow=Nsims,ncol=nhatch)
for(jj in 1:nhatch){
  Ehatchsampled[,jj] <-rbinom(Nsims,size=round(Nhos[jj]),prob=theta)
}
Enatsampled <-rbinom(Nsims,size=round(Nnos),prob=theta)

```

```

Em<-matrix(NA,nrow=Nsims,ncol=nhatch)
Emcwt<-matrix(NA,nrow=Nsims,ncol=nhatch)

for(ii in 1:Nsims){
  for(jj in 1:nhatch){
    Em[ii,jj]<-rbinom(1,size=Ehatchsampled[ii,jj],prob=lambda[jj])
    Emcwt[ii,jj]<-rbinom(1,size=Em[ii,jj],prob=phi[jj])
  }
}

#total unmarked fish (summing over all hatcheries)
Emtot<-apply(Em,c(1),sum)
Eu<-apply(Ehatchsampled,c(1),sum)-Emtot+Enatsampled

Nhoshat<-rep(NA,Nsims)
#Replications of estimates
if(!phitest){
  for(ii in 1:Nsims){
    Nhoshat[ii]<-get.nhoshat(x1=Emcwt[ii,],x2=sum(Em[ii,]-Emcwt[ii,]),theta=lambda,phi=phi)
  }else{
    for(ii in 1:Nsims){
      Nhoshat[ii]<- sum(Emcwt[ii,]/(theta*lambda))
    }
  }

Ntothat<-Eu*(1/theta)+Emtot*(1/theta)
Nnoshat<-Ntothat-Nhoshat
phoshat<-Nhoshat/Ntothat

#properties of phos estimator
SE.Nhoshat<-sqrt(var(Nhoshat,na.rm=T))
CV.Nhoshat<-SE.Nhoshat/sum(Nhos)
SE.Nnoshat<-sqrt(var(Nnoshat,na.rm=T))
CV.Nnoshat<-SE.Nnoshat/Nnos
SE.phoshat<-sqrt(var(phoshat,na.rm=T))
CV.phoshat<-SE.phoshat/phos
BIAS.phoshat<-(mean(phoshat,na.rm=T)-phos)/phos

myres<-list(Nsims=Nsims,
            Nnos=Nnos,
            Nhos=Nhos,
            theta=theta,
            lambda=lambda,
            phi=phi,
            phos=phos,
            SE.Nnoshat=SE.Nnoshat,
            CV.Nnoshat=CV.Nnoshat,

```

```

        SE.Nhoshat=SE.Nhoshat,
        CV.Nhoshat=CV.Nhoshat,
        SE.phoshat=SE.phoshat,
        CV.phoshat=CV.phoshat,
        BIAS.phoshat=BIAS.phoshat)
return(myres)
}

#Theoretical results
phos.mhatch.estimate2<-function(Nnos=200,Nhos=c(100,100),theta=0.25,
    lambda=c(0.75,.25),phi=c(.5,.9)){

#check dimension of inputs
k1<-length(Nhos);k2<-length(lambda);k3<-length(phi)
mytest<-abs(k1-k2)+abs(k2-k3)
if(mytest>0) stop("dimensions of Nhos, lambda, and phi must match")
nhatch<-length(Nhos)

#check inputs
if(sum(Nhos<0))stop("An Nhos estimate is negative")
if(Nnos<0)stop("Nnos estimate is negative")

#check lambdas (if they are all the same, the analysis simplifies)
if(nhatch==1){ mytest==TRUE}
if(nhatch>1){ mytest<-var(lambda)<1.e-10}
if(mytest){
#phis don't matter at all – it's as if there were a single hatchery
res<-phos.estimate2(Nnos=Nnos,
                    Nhos=sum(Nhos),
                    theta=theta,
                    lambda=mean(lambda))
phos=sum(Nhos)/(sum(Nhos)+Nnos)
myres<-list(Nnos=Nnos,
            Nhos=Nhos,
            theta=theta,
            lambda=lambda,
            phi=phi,
            phos=phos,
            SE2.Nnoshat=res$SE2.Nnoshat,
            CV2.Nnoshat=res$CV2.Nnoshat,
            SE2.Nhoshat=res$SE2.Nhoshat,
            CV2.Nhoshat=res$CV2.Nhoshat,
            SE2.phoshat=res$SE2.phoshat,
            CV2.phoshat=res$CV2.phoshat)
return(myres)
}

```

```

}#mytest

#check phis (must all exceed zero)
if(sum(phi==0))stop("phis must all be greater than zero")
phitest<-FALSE
if(sum(phi==1)==nhatch)phitest<-TRUE
phos<-sum(Nhos)/(sum(Nhos)+Nnos)

#theoretical formula for variance of phoshat
Ntot<-sum(Nhos)+Nnos
phosi<-Nhos/Ntot
if(!phitest){
  sum1<-sum(phosi*(1-theta*lambda*phi)/(theta*lambda*phi))
  sum2<-sum(phosi*(1-phi)/phi)
  sum3<-sum(phosi*(1-phi)*theta*lambda/phi)
  phos.var<-(1/Ntot)*(sum1-sum2*sum2/sum3-phos*phos*(1-theta)/theta)
  sum1<-sum(Nhos*(1-theta*lambda*phi)/(theta*lambda*phi))
  sum2<-sum(Nhos*(1-phi)/phi)
  sum3<-sum(Nhos*(1-phi)*theta*lambda/phi)
  Nhos.var<-sum1-sum2*sum2/sum3
  Nnos.var<-Ntot*(1-theta)/theta+Nhos.var-2*(1-theta)*sum(Nhos)/theta
}else{
  sum1<-sum(phosi*(1-theta*lambda)/(theta*lambda))
  phos.var<-(1/Ntot)*(sum1-phos*phos*(1-theta)/theta)
  Nhos.var<-sum(Nhos*(1-theta*lambda)/(theta*lambda))
  Nnos.var<-Ntot*(1-theta)/theta+Nhos.var-2*(1-theta)*sum(Nhos)/theta
}
SE2.phoshat<-sqrt(phos.var)
CV2.phoshat<-SE2.phoshat/phos
SE2.Nhoshat<-sqrt(Nhos.var)
CV2.Nhoshat<-SE2.Nhoshat/sum(Nhos)
SE2.Nnoshat<-sqrt(Nnos.var)
CV2.Nnoshat<-SE2.Nnoshat/Nnos

myres<-list(Nnos=Nnos,
            Nhos=Nhos,
            theta=theta,
            lambda=lambda,
            phi=phi,
            phos=phos,
            SE2.Nnoshat=SE2.Nnoshat,
            CV2.Nnoshat=CV2.Nnoshat,
            SE2.Nhoshat=SE2.Nhoshat,
            CV2.Nhoshat=CV2.Nhoshat,
            SE2.phoshat=SE2.phoshat,

```



```

        CV2.phoshat=CV2.phoshat)
return(myres)
}

```

```

#special case where all lambdas are the same (Bootstrapping Results)
phos.estimates1<-function(Nsims=10000,Nnos=100,Nhos=100,theta=0.25,lambda=0.75)
{
  Ntot<-Nhos+Nnos
  phos<-Nhos/Ntot
  Ehatchsampled <-rbinom(Nsims,size=round(Nhos),prob=theta)
  Enatsampled <-rbinom(Nsims,size=round(Nnos),prob=theta)
  Em<-rep(NA,Nsims)
  for(ii in 1:Nsims){
    Em[ii]<-rbinom(1,size=Ehatchsampled[ii],prob=lambda)
  }
  Eu<-Ehatchsampled-Em+Enatsampled

  Nhoshat<-Em*(1/theta)*(1/lambda)
  Ntothat<-Eu*(1/theta)+Em*(1/theta)
  Nnoshat<-Ntothat-Nhoshat
  phoshat<-Nhoshat/Ntothat
  SE.Nhoshat<-sqrt(var(Nhoshat,na.rm=T))
  CV.Nhoshat<-SE.Nhoshat/Nhos
  SE.Nnoshat<-sqrt(var(Nnoshat,na.rm=T))
  CV.Nnoshat<-SE.Nnoshat/Nnos
  SE.phoshat<-sqrt(var(phoshat,na.rm=T))
  CV.phoshat<-SE.phoshat/phos
  BIAS.phoshat<-(mean(phoshat,na.rm=T)-phos)/phos

  myres<-list(Nsims=Nsims,
              Nnos=Nnos,
              Nhos=Nhos,
              theta=theta,
              lambda=lambda,
              phos=phos,
              SE.Nhoshat=SE.Nhoshat,
              CV.Nhoshat=CV.Nhoshat,
              SE.Nnoshat=SE.Nnoshat,
              CV.Nnoshat=CV.Nnoshat,
              SE.phoshat=SE.phoshat,
              CV.phoshat=CV.phoshat,

```

```

BIAS.phoshat=BIAS.phoshat)

return(myres)
}

#special case where all lambdas are the same (theoretical results)
phos.estimates2<-function(Nnos=100,Nhos=100,theta=0.25,lambda=0.75)
{
  Ntot<-Nhos+Nnos
  phos<-Nhos/Ntot
  var.Nhoshat<-Nhos*(1-lambda*theta)/(lambda*theta)
  var.Nnoshat<-Nnos*(1-theta)/theta+Nhos*(1-lambda)/(theta*lambda)
  var.phos<-(phos/Ntot)*((1-lambda*theta)/(lambda*theta)-phos*(1-theta)/theta)
  SE2.Nhoshat<-sqrt(var.Nhoshat)
  CV2.Nhoshat<-SE2.Nhoshat/Nhos
  SE2.Nnoshat<-sqrt(var.Nnoshat)
  CV2.Nnoshat<-SE2.Nnoshat/Nnos
  SE2.phoshat<-sqrt(var.phos)
  CV2.phoshat<-SE2.phoshat/phos

  myres<-list(Nnos=Nnos,
              Nhos=Nhos,
              theta=theta,
              lambda=lambda,
              phos=phos,
              SE2.Nnoshat=SE2.Nnoshat,
              CV2.Nnoshat=CV2.Nnoshat,
              SE2.Nhoshat=SE2.Nhoshat,
              CV2.Nhoshat=CV2.Nhoshat,
              SE2.phoshat=SE2.phoshat,
              CV2.phoshat=CV2.phoshat)

  return(myres)
}

#In general the estimate depends on the true values
#of escapement, so use iteration until the estimate converges
#Use fixed point iteration to get the GLSE
get.nhoshat<-function(x1,x2,theta,lambda,phi){
  etol<-1.e-10
  nhatch<-length(x1)
  Nhos0<-x1/(theta*lambda*phi)
  if(sum(c(x1,x2))<1.e-10)return(0.0)
  if((sum(x1)<1.e-10)&(x2>0))return(NA)
  run1<-sum(x1*(1-phi)/phi)

```

```

#initial guess
Nhos<- Nhos0
mynorm1<-sqrt(sum(Nhos*Nhos))
err<-2.*etol*(mynorm1+etol)
iter<-0
while(err>etol*(mynorm1+etol)){
  rise<-Nhos*(1-phi)/(phi*theta)
  run<-sum(lambda*Nhos*(1-phi)/phi)
  Nhos<-Nhos0+(rise/run)*(x2-run1)
  mynorm2<-sqrt(sum(Nhos*Nhos))
  err<-abs(mynorm2-mynorm1)
  mynorm1<-mynorm2
  iter<-iter+1
  if(iter>100)stop("too many iterations in get.nhoshat")
}
# print(iter)
return(sum(Nhos))
}

#
get.nhoshat.all<-function(x1,x2,theta,lambda,phi){
  etol<-1.e-10
  nhatch<-length(x1)
  Nhos0<-x1/(theta*lambda*phi)
  if(sum(c(x1,x2))<1.e-10)return(rep(0.0,nhatch))
  if((sum(x1)<1.e-10)&(x2>0))return(rep(NA,nhatch))
  run1<-sum(x1*(1-phi)/phi)
#initial guess
Nhos<- Nhos0
mynorm1<-sqrt(sum(Nhos*Nhos))
err<-2.*etol*(mynorm1+etol)
iter<-0
while(err>etol*(mynorm1+etol)){
  rise<-Nhos*(1-phi)/(phi*theta)
  run<-sum(lambda*Nhos*(1-phi)/phi)
  Nhos<-Nhos0+(rise/run)*(x2-run1)
  mynorm2<-sqrt(sum(Nhos*Nhos))
  err<-abs(mynorm2-mynorm1)
  mynorm1<-mynorm2
  iter<-iter+1
  if(iter>100)stop("too many iterations in get.nhoshat")
}
# print(iter)
return(Nhos)
}

```

Appendix B Names of equivalent variables used in R-code and the mathematical derivations.

Table B.1. —Names of equivalent variables.

R code	Mathematical derivation	Definition
Nhos	H	Total number of hatchery-origin spawners
phos	p	Proportion of hatchery-origin spawners
Nnos	W	Total number of wild-origin spawners
theta	θ	Sampling rate
pcwt	ϕ	CWT fraction
lambda	λ	VM fraction
Ntot	E	Total number of spawners (total escapement)