

The Petersen-Estimator (and much more) for Two-Sample Capture-Recapture Studies with Applications to Fisheries Management

Carl James Schwarz

2025-10-24 19:02:04

Table of contents

1 Installation & Change log	5
1.1 Installation	5
1.2 Access to accompanying workbook	5
1.3 Change log	6
2 Introduction	6
3 Basic Sampling protocols and estimation	7
3.1 Conceptual basis	7
3.2 Sampling protocols	10
3.3 Estimation	11
3.3.1 Maximum likelihood estimation	11
3.3.2 Conditional likelihood estimation	13
3.3.3 Confidence intervals	14
3.4 Software	16
3.5 Example - Estimating the number of fish in Rödli Tarn	17
3.5.1 Petersen estimate	18
3.5.2 Applying a bias correction	19
3.5.3 A model with equal capture probabilities	20
4 Planning studies	21
5 Assumptions and effects of violations	28
5.1 Introduction	28

5.2	Non-closure	28
5.2.1	Immediate handling mortality	28
5.2.2	Natural mortality or emigration	29
5.2.3	Immigration or births	30
5.2.4	Both immigration and mortality	31
5.3	Marking has no effect	31
5.3.1	Marking affects survival	31
5.3.2	Marking affect catchability	32
5.4	Tag loss	33
5.5	Tags not reported or overlooked	33
5.5.1	Inspecting a sub-sample of “un>tagged” fish	34
5.5.2	Reward tags	34
5.6	Homogeneity in catchablity	34
5.6.1	Pure heterogeneity	34
5.6.2	Variable heterogeneity	35
5.7	Spreadsheet to investigate impact of assumption violations	36
5.8	Empirical adjustments to existing estimates	40
5.8.1	Adjusting for tag loss	41
5.8.2	Overlooked tags/Non-reporting of tags	43
5.8.3	Adjusting tags available using Bayesian methods	44
6	Accounting for heterogeneity I - fixed discrete strata	44
6.1	Fixed individual categorical covariates such as sex	45
6.1.1	Test for equal marked fractions	45
6.1.2	Test for equal catchability	45
6.1.3	Stratified models	46
6.2	Example: Northern Pike - simple stratification	47
6.2.1	Summary statistics by sex	47
6.2.2	Test for equal marked fractions by sex	48
6.2.3	Test for equal recapture probabilities by sex	49
6.2.4	Fitting multiple models by sex	50
6.3	Example: Northern pike stratified by length class	54
6.3.1	Distribution of length in captured fish	54
6.3.2	Summary statistics by length class	54
6.3.3	Test for equal marked fractions by length class	56
6.3.4	Test for equal recapture probabilities by length class	56
6.3.5	Fitting multiple models by length class	57
7	Accounting for heterogeneity II - continuous fixed covariate	62
7.1	Quadratic relationship between the probability of capture and length - Northern Pike	63
7.2	Spline relationship between the probability of capture and length - Northern Pike	69
7.3	Non-parametric smoothing	75

8 Accounting for heterogeneity III - geographic or temporal stratification	78
8.1 Introduction	78
8.2 Geographic stratification	80
8.2.1 Sampling protocol	80
8.2.2 Data structure	80
8.2.3 Key assumptions	81
8.2.4 Harrison River Example	81
8.2.5 Fitting the SPAS model	82
8.2.6 Results from model 1 (no pooling).	83
8.2.7 Pooling some rows and columns	87
8.2.8 Not all tagged fish have tags read in recovery strata	99
8.2.9 Summary	99
8.3 Temporal stratification	100
8.3.1 Sampling Protocol	100
8.3.2 Data structure	100
8.3.3 Key assumptions	101
8.3.4 Diagonal Model	102
8.3.5 Non-diagonal case	118
9 Incomplete or partial stratification	124
9.1 Introduction	124
9.2 Sampling protocol	124
9.3 Capture histories and parameters	126
9.4 Example - Walleye in Milles Lac, MN.	126
9.5 Optimal allocation	133
9.6 Additional details	134
10 Double tagging studies to account for tag loss or non-reporting	135
10.1 Introduction	135
10.2 Capture histories	137
10.3 Example - Kookanee on Metolius River - two tags that cannot be distinguished	139
10.4 Example - Simulated data - two tags that can be distinguished	141
10.4.1 Model with equal tag retention probabilities	142
10.4.2 Model with unequal tag retention probabilities	142
10.4.3 Comparing models and model averaged estimates of abundance	143
10.5 Example - Simulated data - a permanent second tag used	144
10.6 Example - Northern Pike - tag loss	145
10.7 Example - Simulated data - non-reporting of tags and reward tags	149
10.8 Planning tag loss studies	150
10.8.1 Example of deciding between proportion of double tagging	151
11 Multiple marks	152
11.1 Introduction	152

11.2 Theory	153
11.3 Illustration of using multimark with simulated data.	156
11.3.1 LR-type surveys	156
11.3.2 LRB-type surveys	168
11.4 Bobcat example	175
12 Multiple Petersen estimates	184
12.1 Introduction	184
12.2 Batch marks (logit-normal model)	185
12.3 Example of PNE method - TBA	193
12.4 Summary	193
13 Forward and reverse capture-recapture studies	193
13.1 Introduction	193
13.2 Example - Run size of Yukon River Chinook	194
13.3 Example: Lower Fraser Coho - SPAS in reverse	196
13.3.1 Fitting the SPAS model	198
13.4 Example combining forward and reverse capture-recapture studies	200
13.5 Summary	200
14 Genetic capture-recapture methods using close kin studies	200
14.1 Transgenerational capture-recapture	201
14.1.1 Introduction	201
14.1.2 Bailey Binomial model	202
14.1.3 Hypergeometric model	204
14.1.4 Caution on using simple sample methods.	204
14.1.5 Limitations and Assumptions	205
15 Grand summary	206
16 Miscellaneous notes	208
16.1 Speeding computation times	208
17 References	209
18 Appendix A - The multinomial models for the Petersen study	214
18.1 Complete multinomial model	214
18.2 Conditional multinomial model	216
19 Appendix B - Using the VGAM package	218
19.1 Data structure	218
19.2 Example Rodli Tarn	218
19.3 Example Rodli Tarn - Chapman correction	220
19.4 Example - Rodli Tarn - Equal capture probabilities	221

19.5 Example - Northern Pike - Stratification by Sex	223
19.6 Example - Northern Pike - Stratification by Length Class	224
20 Appendix C. Using RMark/MARK	225
20.1 Data structure	225
20.2 Example Rodli Tarn	225
20.3 Example Rodli Tarn - Chapman correction	227
20.4 Example - Rodli Tarn - Equal capture probabilities	229
20.5 Example - Northern Pike - Stratification by Sex	230
20.6 Example - Northern Pike - Model with length as a covariate	233
21 Appendix D. Conditional likelihood approach to tag-loss studies.	237
22 Appendix E: Brief theory for BTSPAS models.	238
22.1 Diagonal case	238
22.2 Non-diagonal case	241

1 Installation & Change log

1.1 Installation

The *Petersen* package can be installed from CRAN in the usual way.

The development version can be installed from within *R* using:

```
devtools::install_github("cschwarz-stat-sfu-ca/Petersen",
                        dependencies = TRUE,
                        build_vignettes = TRUE)
```

1.2 Access to accompanying workbook

The accompanying workbook is available in the external data directory of the package and can be copied to your working directory using:

```
extdata.dir <- system.file("extdata", package="Petersen")
extdata.dir
```

```
[1] "/Users/cschwarz/Rlibs/Petersen/extdata"
```

```

extdata.list <- list.files(extdata.dir)
extdata.list

[1] "PetersenWorkBook-2023-05-01.xls"

# Uncomment the following to copy the PetersenWorkbook to your working directory.
#file.copy(from=file.path(extdata.dir, "PetersenWorkbook-2006-01-20.xls"), to=getwd())

```

1.3 Change log

Date	Changes
2025-03-01	Added n1_n2_m2_to_cap_hist() function to generate capture histories summary statistics
2024-06-01	Added new example of forward-reverse capture-recapture (lower Fraser coho)
	Added chapter on use of multimark when you have left and right photographs and cannot match left and right to the same animal.
2023-12-01	Updates for CRAN submission. No new functionality or examples.
2023-05-01	First Release

2 Introduction

In the late 1890's, Petersen (1896) estimated exploitation probabilities and the abundance of fish living in enclosed bodies of water. His methods have been widely adopted and the Petersen-method is among the most widely used methods in fisheries management.

Similar methods were used by Laplace (1793) to estimate the population of France based on birth registries; by Lincoln (1930) to estimate the abundance of ducks from band returns; and by Jackson (1933) to estimate the density of tsetse flies.

The Petersen-method is the simplest of more general capture-recapture methods which are extensively reviewed in Williams et al. (2002). Despite the Petersen method's simplicity, many of the properties of the estimator, and the effects of violations of assumptions are similar to these more complex capture-recapture studies. Consequently, a firm understanding of the basic principles learned from studying this method are extremely useful to develop an intuitive understanding of the larger class of studies.

The purpose of this monograph is to bring together a wide body of older and newer literature on the design and analysis of the “simple” two-sample capture-recapture study. This monograph builds upon the comprehensive summaries found in Ricker (1975), Seber (1982), Seber and Schofield (2023), and William et al (2002), and incorporates newer works that have not yet summarized. While the primary emphasis is on the application to fisheries management, the methods are directly applicable to many other situations.

Computer software has revolutionized many aspects of data analysis. A workbook accompanies this monograph to assist in the design and analysis of the Petersen studies. As well, the *Petersen* package in *R* is available for download.

Following Stigler’s law of eponymy https://en.wikipedia.org/wiki/Stigler%27s_law_of_eponymy, Goudie and Goudie (2007) investigated the origin of the Petersen estimator and identified additional scientists whose early work on marking and estimating fish populations deserves more credit than it has received.

3 Basic Sampling protocols and estimation

3.1 Conceptual basis

The fundamental goal of a Petersen study is to estimate N , the number of animals (the abundance) in the population of interest.

The idealized protocol consists of an initial capture of n_1 animals from the population. These animals are given a mark or tag (Figure 1).

Marks or tags fall into two general categories. Batch marks are simple marks that only identify that an animal was captured at a particular occasion. It is impossible to identify individual animals from batch marks. While batch marking is sufficient for the Petersen and other two-sample methods, modern practice is to use individually numbered tags so that the capture history of each individual animal can be determined and individual animals can be identified from each other.

The marked/tagged animals are then returned to the population. After allowing the marked and remaining unmarked animals to mix, a second sample of size n_2 is selected. Each animal in the second sample is examined, and a count of the number of animals marked in the first sample and now recaptured, m_2 is taken.

The summary statistics n_1 , n_2 , and m_2 are sufficient statistics for this study. Modern practice is to record information in terms of capture histories rather than these summary statistics. A capture history, ω is a vector where component i of the vector takes the value of 1 if an animal was captured at sampling event (time) i and the value 0 if the animal was not captured at the sampling event (time) i . Because the Petersen study only has two capture occasions, all capture histories are of length 2 where:

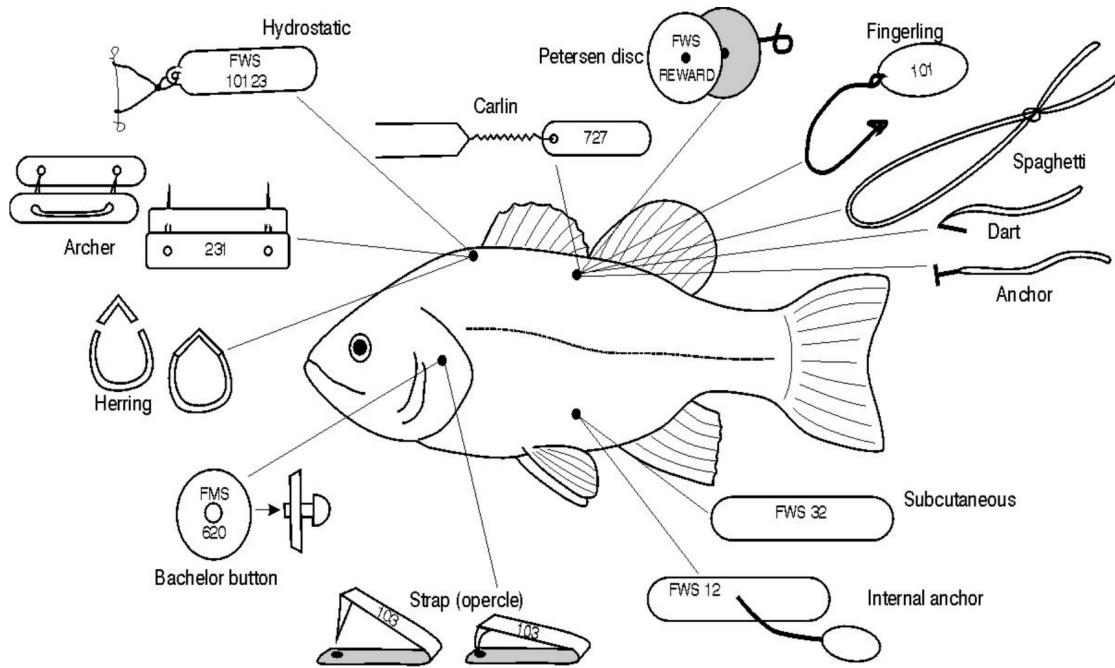


Figure 4.1 Principal types of external fish tags and attachment sites (redrawn from McFarlane *et al.*, 1990).

Figure 1: Common tag types and tagging locations. Diagram taken from XXXX.

- $\omega = \{1, 1\}$ represents an animal captured at both sampling occasions.
- $\omega = \{1, 0\}$ represents an animal captured at the first occasion but not at the second occasion.
- $\omega = \{0, 1\}$ represents an animal not captured at the first occasion but captured at the second occasion.
- $\omega = \{0, 0\}$ represents animals not captured at either sampling occasion (not observable).

The notation n_ω represents the number of animals with capture history ω . Note that

- $N = n_{\{0,0\}} + n_{\{0,1\}} + n_{\{1,0\}} + n_{\{1,1\}},$
- $n_1 = n_{\{1,0\}} + n_{\{1,1\}},$
- $n_2 = n_{\{0,1\}} + n_{\{1,1\}}$ and
- $m_2 = n_{\{1,1\}}.$

The capture history notation is especially convenient for more complex capture-recapture studies. Both notations will be used in this monograph.

The fundamental estimating equation is based on the idea that the proportion of marked animals in the second sample should be approximately equal to the proportion of animals initially captured:

$$\frac{m_2}{n_2} \approx \frac{n_1}{N}$$

By rearranging this relation, the Petersen estimator (Equation 1) is obtained

$$\widehat{N}_{Petersen} = \frac{n_1 n_2}{m_2} \tag{1}$$

The estimated capture probabilities at each sample occasion can also be obtained:

$$\begin{aligned}\hat{p}_1 &= \frac{n_1}{\widehat{N}} = \frac{m_2}{n_2} \\ \hat{p}_2 &= \frac{n_2}{\widehat{N}} = \frac{m_2}{n_1}\end{aligned}$$

If \widehat{N} is to be a sensible estimator of N , several assumptions must be made. The effects of violations of these assumptions will be discussed in Section 5.

Following Seber (1982) these assumptions are:

1. **Closure.** In other words, no animals leave the population (by death or emigration) and no animals enter the population (by internal births or immigration) between sampling occasions.

2. **Homogeneity.** This assumption can be satisfied in a number of ways. All animals have the same probability of capture at the first occasion, or all animals have the same probability of capture at the second occasion, or the second sample is a simple random sample selected from the entire population, i.e., each animal is captured or not captured independently of every other animal with the same probability. This assumption allows for some latitude in the study. For example, fish may be conveniently marked in the first sample without worrying about randomization if the scientist is sure that the second sample is a random sample from the entire population. Of course, it is likely wise to make such a strong assumption and randomization in each sample is highly recommended.
3. **No impact of marking.** Marking the animal does not affect its subsequent survival or catchability, i.e., all animals in the second sample, regardless of marking status, have the same probability of capture.
4. **No tag loss.** No marked animal loses its mark between the two sampling occasions.
5. **No missed tags.** All marked animals are correctly identified in the second sample.

3.2 Sampling protocols

While the estimator \widehat{N} in Equation 1 is intuitively appealing, its properties (bias and precision) cannot be investigated unless the sampling scheme that results in the observed data is fully specified. There are many sampling protocols, the most common being:

- **Direct Sampling.** In direct sampling method the sample sizes (n_1 and n_2) are specified in advance.}
- **Inverse sampling.** In Inverse Sampling, sampling continues until a certain number of marks is obtained.

For each of these protocols, fish can be sampled with or without replacement giving:

- **Hypergeometric Sampling.** In this method sampling occurs without replacement;
- **Binomial sampling.** In the binomial model (sampling with replacement); and

Modern statistical methods for capture-recapture data consider the number of animals in each capture history to be a realization of a multinomial distribution, which is a generalization of the binomial sampling methods.

3.3 Estimation

3.3.1 Maximum likelihood estimation

Once the sampling model has been established, the standard method of obtaining the estimator is Maximum Likelihood Estimation (MLE). The likelihood equations for the various sampling protocols is shown in Table 1.

Maximum Likelihood Estimators asymptotically unbiased and fully efficient (i.e., make use of all of the data and results in the smallest possible standard error). It turns out that for all the sampling schemes discussed above that the estimator \widehat{N} from Equation 1 is the MLE.

While the MLE is optimal in large samples, it can be severely biased with smaller sample sizes because of the presence of m_2 in the denominator. Indeed, when $E[m_2]$ is small, there is a fairly large probability that no marks would be observed (i.e., $m_2 = 0$) and $\widehat{N} = \infty$. Chapman (1951) suggested a simple modification to the estimator as shown in Table 1. **Add reference here to Rivest's work on determining the optimal correction factor.**

These modifications removes most of the bias, and Robson and Regier (1964) showed that the approximate residual bias (when $n_1 + n_2 < N$) is

$$b = E \left[\frac{\widehat{N}_{HU} - N}{N} \right] = E \left[-\exp \left(-\frac{(n_1 + 1)(n_2 + 1)}{N} \right) \right]$$

If $E[m_2] > 4$, the residual relative bias is less than 2% of the abundance. To allow for variation in the observed m_2 around its expected value, Robson and Regier (1964) recommend that studies have $m_2 > 7$ to be 95% confident that the $E[m_2] > 4$ and that the residual relative bias in \widehat{N}_{HU} is negligible. This is equivalent to ensuring that $n_1 n_2 > 4N$ as outlined by Robson and Regier (1964).

Table 1: Summary of sampling models, bias adjusted estimators, and estimated variance of estimators

Sampling Model	Likelihood	Bias adjusted estimator	Estimated variance
Direct	$\frac{\binom{n_1}{m_2} \binom{N-n_1}{n_2-m_2}}{\binom{N}{n_2}}$	$\widehat{N}_{HU} = \frac{(n_1+1)(n_2+1)}{(m_2+1)} - 1$	$\widehat{v}_{HU} = \frac{(n_1+1)(n_2+1)(n_1-m_2)(n_2-m_2)}{(m_2+1)^2(m_2+2)}$
Hypergeometric			
Binomial	$\binom{n_2}{m_2} \left(\frac{n_1}{N}\right)^{m_2} \left(1 - \frac{n_1}{N}\right)^{n_2-m_2}$	$\widehat{N}_{BU} = \frac{n_1(n_2+1)}{m_2+1}$	$\widehat{v}_{BU} = \frac{n_1^2(n_2+1)(n_2-m_2)}{(m_2+1)^2(m_2+2)}$

Sampling Model	Likelihood	Bias adjusted estimator	Estimated variance
Inverse Hypergeometric	$\frac{\binom{n_1}{m_2-1} \binom{N-n_1}{n_2-m_2}}{\binom{N}{n_2-1}} \times \frac{n_1-m_2+1}{N-n_2+1}$	$\widehat{N}_{IHU} = \frac{(n_1+1)n_2}{m_2} - 1$	to be added later
Inverse Binomial	$\binom{n_2-1}{m_2-1} \left(\frac{n_1}{N}\right)^{m_2} \left(1 - \frac{n_1}{N}\right)^{n_2-m_2} \widehat{N}_{IBU} = \frac{n_1 n_2}{m_2}$		$\widehat{v}_{IBU} = \frac{n_1^2 n_2 (n_2 - m_2)}{m_2^2 (m_2 + 1)}$
Capture History Multinomial	$\binom{N}{n_{00}, n_{01}, n_{10}, n_{11}} \times ((1-p_1)(1-p_2))^{n_{00}} \times (p_1(1-p_2))^{n_{10}} \times ((1-p_1)p_2)^{n_{01}} \times (p_1 p_2)^{n_{11}}$	$\widehat{N}_{MN} = \frac{(n_{11}+n_{10})(n_{11}+n_{01})}{n_{11}} = \frac{n_1 n_2}{m_2}$	$\widehat{v}_{MN} = \frac{n_1 n_2}{m_2} \frac{(n_2 - m_2)}{m_2} \frac{(n_1 - m_2)}{m_2}$

Chapman (1951) also derived the variance of the hypergeometric estimator and expressed it as:

$$V(\widehat{N}_{HU}) = N^2 (E[m_2]^{-1} + 2E[m_2]^{-2} + 6E[m_2]^{-3})$$

If $E[m_2]$ is small, then the variance is large. For example if $E[m_2] = 10$, then the relative standard error ($RSE = \frac{SE}{estimate}$) is over 35% and a 95% confidence interval will only be accurate to within 70% of the true value of N ! A discussion of sample size requirements is presented in Section 4.

In cases where the marked fraction is very small (i.e., $n_1 \ll N$) or when sampling is done with replacement (e.g. when animals are observed rather than physically captured), the binomial model of Bailey (1951, 1952) may be used as summarized in Table 1. Seber (1982, p. 61) also points out that the Bailey estimator may be appropriate when complete mixing of marked and unmarked animals takes places and a systematic sample rather than a simple random sample is taken. The Bailey adjusted estimator has comparable residual bias to the Chapman adjusted estimator and is not reported here. In most practical situations, there is little difference between the hypergeometric and binomial sampling model estimates or estimated variances.

In inverse sampling, the number of recaptures (m_2) to be obtained is fixed in advance and the size of the second sample (n_2) is now random. This was considered by Bailey (1951), Chapman (1951), Robson (1979); summarized by Seber (1982, Section 3.8); and results are presented in Table 1. Both inverse sampling methods are more efficient (i.e., for a given precision of the estimated abundance, the expected sample size required to obtain this precision under inverse sampling is smaller than that required under direct sampling) than their direct sampling counterparts, but the gain in efficiency is, in practice, negligible. The primary disadvantage of inverse sampling is that with poor planning, the expected sample size could be very large.

Seber (1982) summarizes an alternate sampling scheme in which the number of unmarked individuals captured in the second sample is fixed.

Seber (1982) discusses the case of random sample sizes and notes that in practice one would condition upon the observed values so that these previous models are still appropriate. One can also think of a model where the number of fish in the four possible tag histories is random which naturally evolves into more complex studies with multi-samples from both open and closed populations.

In the multinomial model, the counts in the four possible capture histories is considered to arrive from a multinomial model. This multinomial model is the predominant paradigm in current capture-recapture methodology and will be used in the remainder of this monograph. The derivation of the results under the multinomial sampling protocol is found in Section 18.1 and the results are also summarized in Table 1.

In some cases, despite best efforts, no recapture are observed, i.e., $m_2 = 0$. While the MLE is nonsensical, the adjusted estimators of Table 1 do provide “estimates” but these will be of very poor precision. Alternatively, Bell (1974) showed that under the hypergeometric sampling model

$$P(m_2 = 0) = \frac{(N - n_1)!(N - m_2)!}{N!(N - n_1 - n_2)!}$$

and suggested solving $P(m_2 = 0) = .5$ as an estimator of N with the solutions to $P(m_2 = 0) = .025$ and $P(m_2 = 0) = .975$ as 95% confidence bounds for N . While this works in theory, the resulting estimator has such poor precision that the practical application of this results is doubtful.

Finally, it should be noted, that most sampling plan do not fall exactly into any of the above categories. In many practical plans, the amount of effort (e.g. person-days) is specified for both the initial and second sample, and the number of fish that are captured depends on chance events within that period of effort and should be considered random.

In most studies, results from any of the estimators are similar enough that the practitioner should not be too concerned about the choice of appropriate model – error in the estimates attributable to other problems in the experiment such as assumptions not being satisfied exactly will likely be an order of magnitude greater than any differences in estimates of abundance or precision among the model formulations.

3.3.2 Conditional likelihood estimation

While the MLE are fully efficient, an alternate estimator can be derived that conditions on the observed fish only. Huggins (1989) provides the theory for the conditional multinomial which uses only the observed fish to estimate catchabilities and abundance is estimated using a Horvitz-Thompson type estimator (Section 18.2)

Its main usage is when catchability depends upon individual covariates (e.g., fish length) which is unknown for fish never caught.

In the case of no individual covariates, the conditional likelihood estimators reduce to the full likelihood estimators with no loss of efficiency (i.e., same standard errors), so in practice, the conditional likelihood approach can always be used and is the basis for estimators found in the *Petersen* package. See Section 18.2 for more details.

3.3.3 Confidence intervals

Confidence intervals for the abundance can be computed in a number of ways.

3.3.3.1 Large sample Wald interval

This method relies upon the asymptotic normality of \widehat{N} and the usual large-sample, asymptotic, normal-theory based confidence interval is:

$$\widehat{N} \pm z_{\alpha/2} SE_{\widehat{N}}$$

However, this interval is not recommended for standard usage for two reasons:

- The distribution of \widehat{N} is typically skewed with a long right tail;
- There is a very strong positive correlation between \widehat{N} and the estimated SE . This implies that when \widehat{N} is below the true population value, the SE tends to also be smaller than the true SE and the resulting confidence interval is too narrow, and vice versa.

Many authors have suggested modifications to improve upon the large-sample asymptotic result above.

3.3.3.2 Logarithmic transform

Programs such as *MARK* compute a confidence interval on the logarithm of \widehat{N} and then invert the corresponding interval.

$$\begin{aligned}\widehat{\theta} &= \log \widehat{N} \\ se_{\widehat{\theta}} &= \frac{se_{\widehat{N}}}{\widehat{N}}\end{aligned}$$

The confidence interval for $\log \widehat{N}$ is found as

$$\widehat{\theta} \pm z_{\alpha/2} se_{\widehat{\theta}}$$

and then the confidence interval for \widehat{N} is found by re-inverting the above interval

$$\exp(\widehat{\theta} - z_{\alpha/2} se_{\widehat{\theta}}) \text{ to } \exp(\widehat{\theta} + z_{\alpha/2} se_{\widehat{\theta}})$$

3.3.3.3 Inverse transformation

Ricker (1975) suggest first using a simple inverse transform, then finding a confidence interval based upon $1/\widehat{N}$, and then re-inverting the resulting confidence interval, i.e., first compute:

$$\begin{aligned}\hat{\theta} &= \frac{1}{\widehat{N}} \\ se_{\hat{\theta}} &= \frac{se_{\widehat{N}}}{\widehat{N}^2}\end{aligned}$$

Then find a confidence interval for $1/N$ based upon the transformed values,

$$\hat{\theta} \pm z_{\alpha/2} se_{\hat{\theta}}$$

and finally then the confidence interval for \widehat{N} is found by re-inverting the above interval

$$\frac{1}{\hat{\theta} + z_{\alpha/2} se_{\hat{\theta}}} \text{ to } \frac{1}{\hat{\theta} - z_{\alpha/2} se_{\hat{\theta}}}$$

3.3.3.4 Inverse cube-root transform

The inverse approximation above appears to work fairly well, but Sprott (1981) used likelihood theory to show that the inverse-cube-root transform more effectively captured the skewness in the sampling distribution. His procedure is very similar to the above:

$$\begin{aligned}\hat{\theta} &= \frac{1}{\widehat{N}^{1/3}} = \frac{1}{\sqrt[3]{\widehat{N}}} \\ se_{\hat{\theta}} &= \frac{se_{\widehat{N}}}{3\widehat{N}^{4/3}}\end{aligned}$$

The confidence interval for $1/N^{1/3}$ is found as

$$\hat{\theta} \pm z_{\alpha/2} se_{\hat{\theta}}$$

and then then the confidence interval for \widehat{N} is found by re-inverting the above interval

$$\frac{1}{(\hat{\theta} + z_{\alpha/2} se_{\hat{\theta}})^3} \text{ to } \frac{1}{(\hat{\theta} - z_{\alpha/2} se_{\hat{\theta}})^3}$$

3.3.3.5 Bootstrapping

Bootstrapping is easy to implement when data is collected as capture-histories (rather than the summary statistics) because each fish is represented by an individual capture history. A resampling of the capture-histories then represents a bootstrap sample; estimates can be computed for each bootstrap sample, and the usually methods for determining confidence intervals from bootstrap samples can be used.

3.3.3.6 Bayesian credible intervals

Add more details later

3.3.3.7 Profile intervals

While the transformation methods have the advantage of simplicity with the computations easily done by hand, a likelihood method uses the likelihood function directly to capture the skewness. Standard likelihood theory shows that the **profile confidence interval** is found by finding all values of N such that:

$$-2l(N) - 2l(\widehat{N}) \leq \chi_{\alpha}^2$$

However, in the conditional likelihood methods. N does NOT appear directly in the likelihood and so these methods are not relevant

3.3.3.8 Which method to choose?

All transformation methods should give similar results when the number of marks returned is reasonable. The methods could give quite different answers in cases when only a few marks are returned (say less than 20). However, in these cases, the differences in results among the confidence interval methods is small relative to the (unknown) biases and uncertainties from assumptions failures (such as non-mixing) that are still resident in the study.

The *Petersen* package created for this monograph used the logarithmic transformation to compute the confidence intervals for N .

It should be noted that confidence intervals only capture the uncertainty in the estimate due to sampling – it does NOT capture uncertainty due to failure of assumptions, problems in the study, or other problems with the study.

3.4 Software

A simple internet search will find much software that can compute the Petersen estimator because the computations are so simple. However, these packages seldom take a coherent modeling strategy for the more complex models for Petersen studies. Similarly, the data structures seldom are consistent with modern software for capture-recapture studies (e.g., MARK).

Consequently, the *Petersen R* package has been developed to use a consistent methodology (conditional maximum likelihood estimation) with a consistent data frame work (capture histories). This package is available from the usual *R* repositories.

The *VGAM* (Yee et al., 2015) package also adopts the conditional likelihood approach for the closed population models with 2+ sample times, of which Petersen studies are simple cases. The use of this package is illustrated in Section 19.

The *MARK* program (and *RMark* which calls *MARK*) could be used for the simple Petersen studies, including the conditional likelihood approach. The use of this package is illustrated in Section 20.

3.5 Example - Estimating the number of fish in Rödli Tarn

Ricker (1975) gives an example of work by Knut Dahl on estimating the number of brown trout (*Salmo trutta*) in some small Norwegian tarns. Between 100 and 200 trout were caught by seining, marked by removing a fin (an example of a batch mark) and distributed in a systematic fashion around the tarn to encourage mixing. A total of $n_1 = 109$ fish were captured, clipped and released, $n_2 = 177$ fish were captured at the second occasion, and $m_2 = 57$ marked fish were recovered.

The data are available in the *data_rodli* data frame in the *Petersen* package and can be accessed in the usual fashion:

```
library(Petersen)
data(data_rodli)

data_rodli
##   cap_hist freq
## 1      11    57
## 2      10    52
## 3      01   120
```

The data frame consists of (at least) two columns:

- a variable *cap_hist* with the two digit capture history (a character vector) -
- a variable *freq* that contains the number of fish in each capture history.

Additional columns can be present in the data frame for attributes of the fish such as sex, length etc.

The data frame can have a separate row for each fish (each with *freq=1*), or a summary as shown above.

You can construct the capture histories from the summary statistics as shown below:

```
cap_hist <- Petersen::n1_n2_m2_to_cap_hist( n1=109, n2=177, m2=57)
cap_hist
```

```

cap_hist freq
1      10   52
2      01  120
3      11   57

```

3.5.1 Petersen estimate

We find the Petersen estimator of abundance using the conditional likelihood approach used in the *Petersen* package in two steps (the reason for the two steps will be explained later):

In the first step, the conditional-likelihood model is fit to the data:

```
rodli.fit.mt <- Petersen::LP_fit(data_rodli, p_model=~..time)
```

The *LP_fit()* function takes the data frame of capture histories seen earlier and a model for the capture probabilities (the *p_model* argument). The model for the capture probabilities can refer to any variable in the data frame (e.g., sex) or to several special variables such as *..time* which refer to the two time periods. Some knowledge of how *R* sets up the *design matrix* given the data frame is helpful in deciding how to specify *p_model* in more complex situations involving stratification and is discussed later.

In the standard Petersen estimator, we allow the capture probabilities to vary across the two sampling events. Consequently, the *p_model* was specified as *p_model*=*~..time*.

A summary of the fit is given in a data frame

```

rodli.fit.mt$summary
##   p_model name_model    cond.ll n parms nobs   method
## 1 ~..time p: ~..time -233.9047      2 229 cond ll

```

The value of the conditional likelihood, number of parameters in the conditional likelihood (just the two capture probabilities) and the number of observed fish (sum of the *freq* column) are also presented. Notice that in the conditional likelihood approach, the abundance is NOT a parameter in the likelihood and so is not counted in the model summary table.

In the second step, we obtain estimates of overall abundance using the *LP_est()* function and the *N_hat* argument. Here there is no stratification or other groupings of the data, so the formula for *N_hat* is *~1* indicating that we should find the estimated abundance for the entire population. (In later sections, we will obtain abundance estimates for individual strata as well).

```
rodli.est.mt <- Petersen::LP_est(rodli.fit.mt, N_hat=~1)
```

This again gives a data frame with the estimated abundance.

```
rodli.est.mt$summary
##   N_hat_f     N_hat_rn     N_hat N_hat_SE N_hat_conf_level N_hat_conf_method
## 1      ~1 (Intercept) 338.4737 25.49646          0.95           logN
##   N_hat_LCL N_hat_UCL p_model name_model cond.ll n parms nobs method
## 1 292.0154 392.3232 ~..time p: ~..time -233.9047      2 229 CondLik
```

The estimated abundance is 338 (SE 25) fish computed as:

$$\widehat{N} = \frac{109 \times 177}{57} = 338.$$

The 95% confidence interval for N was computed using a logarithmic transformation is 292 to 392 fish.

The relative standard error (RSE) is found as

$$RSE = \frac{SE}{estimate} = \frac{25.5}{338.5} = .075$$

is similar to the approximation that

$$RSE \approx \frac{1}{\sqrt{(marks\ recovered)}} = \frac{1}{\sqrt{57}} = .13$$

While the process of specifying a model for the capture probabilities and for the abundance estimate may seem a bit convoluted, its real power will become apparent when more complex cases involving stratification are discussed later.

3.5.2 Applying a bias correction

The Chapman modifications can be applied as outlined in Table 1. This can be implemented by adding a single “new” fish with capture history “11” to the existing data frame.

```
data(data_rodli)
rodli.chapman <- plyr::rbind.fill(data_rodli,
                                      data.frame(cap_hist="11",
                                                 freq=1,
                                                 comment="Added for Chapman"))

rodli.chapman
##   cap_hist freq           comment
## 1      11    57           <NA>
```

```

## 2      10   52          <NA>
## 3      01   120         <NA>
## 4      11   1 Added for Chapman

```

Then this adjusted data is passed to the *Petersen* package and the two step procedure is again followed:

```

rodli.fit.mt.chap <- Petersen::LP_fit(rodli.chapman, p_model=~..time)
rodli.est.mt.chap <- Petersen::LP_est(rodli.fit.mt.chap, N=~1)

```

```

rodli.est.mt.chap$summary
##   N_hat_f   N_hat_rn   N_hat N_hat_SE N_hat_conf_level N_hat_conf_method
## 1 ~1 (Intercept) 337.5861 25.02396           0.95             logN
##   N_hat_LCL N_hat_UCL p_model name_model cond.ll n parms nobs method
## 1 291.9364 390.374 ~..time p: ~..time -235.2889       2 230 CondLik

```

The bias-corrected estimate of the abundance is 338 (SE 25) fish.

3.5.3 A model with equal capture probabilities

It is also possible to fit a model where the capture probabilities are equal at both sampling events. This is specified by changing the specification for the model for p from $p = \sim ..time$ to $p = \sim 1$:

```

rodli.fit.m0 <- Petersen::LP_fit(data_rodli, p_model=~1)
rodli.est.m0 <- Petersen::LP_est(rodli.fit.m0, N_hat=~1)

```

This gives the estimates:

```

rodli.est.m0$summary
##   N_hat_f   N_hat_rn   N_hat N_hat_SE N_hat_conf_level N_hat_conf_method
## 1 ~1 (Intercept) 358.7542 28.57733           0.95             logN
##   N_hat_LCL N_hat_UCL p_model name_model cond.ll n parms nobs method
## 1 306.8971 419.3738 ~1           p: ~1 -247.7207       1 229 CondLik

```

Now the estimated abundance is 359 (SE 29) fish.

However, a comparison of the model fits using AICc (Table 2), shows that this model has much less support than the traditional estimator with unequal capture probabilities:

```

rodli.aictab <- Petersen::LP_AICc(
  rodli.fit.mt,
  rodli.fit.m0)

```

Table 2: Model comparison for the Rodli dataset

Model specification	Conditi- onal log- likelihoood	# parms	# obs	AICc	Delta	AICcWt
p: ~..time	-233.9047	2	229	471.86	0.00	1.00
p: ~1	-247.7207	1	229	497.46	25.60	0.00

Notice the N is not part of the conditional likelihood and so is not counted as a parameter.

In general, models with equal capture probability over sampling events are seldom sensible from a biological perspective.

4 Planning studies

Before determining the amount and distribution of effort to be allocated in a capture-recapture study, one must first decide what level of precision is required from the study. In many cases, a desired measure of the relative precision (relative standard error) is a goal of the study.

Some papers use the term *coefficient of variation (cv)* to refer to the relative standard error. The term relative standard error is preferred and the term *cv* is reserved for describing the variability of individual data values, i.e., $cv = \frac{std\ dev}{mean}$.

In turn, the relative standard error can be used to express the relative 95% confidence interval using the approximate rule that the relative confidence interval is about ± 2 relative standard errors.

Seber (1982, p. 64) gives three target levels of precision that are commonly used:

- For preliminary surveys where only a rough idea of the abundance is needed, the relative 95% confidence interval should be $\pm 50\%$ of the estimated abundance with a corresponding relative standard error of 25%. For example, it would be sufficient for a preliminary survey to have results with an estimate of 50 (*SE* 12) thousand fish with a 95% confidence interval of between 25 and 75 thousand fish.

- For accurate management work, the relative 95% confidence interval should be $\pm 25\%$ of the estimated abundance with a corresponding relative standard error of 12.5%. For example, it would be sufficient for accurate management work to have results with an estimate of 50 ($SE \approx 6$) thousand fish, with a 95% confidence interval between 38 and 62 thousand fish.
- For careful scientific research, the recommended relative confidence interval is $\pm 10\%$ of the estimated abundance with a corresponding relative standard error of 5%. For example, it would be sufficient for scientific work to have results with an estimate of 50 ($SE \approx 2.5$) thousand fish, with a 95% confidence interval of between 45 and 55 thousand fish.

If \widehat{N} is to be used for further computations, e.g. splitting the population by size categories or multiplied by average biomass to obtain an estimate of total biomass, then a high degree of precision is required so that the precision of the final answer is adequate.

As a rough rule of thumb, the relative precision (relative standard error) of the Petersen estimate is proportional to $\frac{1}{\sqrt{m_2}}$, i.e., to the square root of the number of marks returned. This can be used for planning purposes. For example, if the initial abundance is about 50,000 fish, and if 5,000 fish are marked and 1,000 fish are examined for marks, then the approximate number of marks recaptured is:

$$E[m_2] \approx 5,000 \times \frac{1,000}{50,000} = 100$$

and the approximate relative standard error will be on the order of

$$rse \approx \frac{1}{\sqrt{100}} = .1$$

This will give a relative 95% confidence interval of about $\pm 20\%$ which should be precise enough for management purposes. In general, the three levels of precision will require a certain number of marks returned as summarized in Table 3:

Table 3: Suggested minimum number of marks required to be returned to meet precision criteria

	Preliminary	Management	Scientific
95% relative ci	$\pm 50\%$	$\pm 25\%$	$\pm 10\%$
Relative SE	25%	12%	5%
Required	16	64	400
$E[m_2]$			

The rule of thumb can be inverted to estimate the approximate fraction of the population that needs to be marked assuming equal sample sizes at both sampling occasions. In this case,

$E[m_2] \approx n^2/N \approx Nf^2$, where f is the sampling fraction. Consequently, if scientific precision is needed (a relative standard error of 5% with 400 marks needed to be recaptured), then a population of 10,000 will require $10,000f^2 = 400$ or $f = .2 = 20\%$ of the population will be needed to be sampled on both occasions.

While these rules of thumb should be sufficient for most purposes, one could actually use the estimated variances presented earlier. The workbook that accompanies this monograph has a worksheet where the scientist can enter various values of N , n_1 , and n_2 to see the approximate expected precision that will be obtained. For example, Figure 2, shows that for a abundance of about $N = 50,000$, sample sizes of $n_1 = 3,000$ and $n_2 = 1,000$ would give a relative standard error of about 12% and a relative 95% confidence interval of about $\pm 25\%$ which should be sufficient for management purposes.

Robson and Regier (1964) gave nomograms which have been adopted and are presented in this monograph in Figure 3 to Figure 5. For example, using Figure 4 for a abundance of $N = 50,000$, the following pairs of sample sizes will be required to have the 95% relative confidence interval to be within $\pm 25\%$:

$$(1,000; 3,000), (800; 4,000), (300; 10,000)$$

The charts are symmetric in n_1 and n_2 , so that the same precision is obtained for $N = 50,000$ with $(n_1 = 1,000; n_2 = 3,000)$ fish and $(n_1 = 3,000; n_2 = 1,000)$ fish.

While Robson and Regier (1964) and Seber (1982) have nomograms for abundances less than 200, these are rarely useful in practice.

The above spreadsheets and nomograms are sufficient for most planning purposes, but it should be kept in mind that these computations assume that the study will proceed perfectly and that all assumptions will be satisfied. This is rarely the case in most surveys, and so the required effort should be increased to account for potential problems down the road.

The above charts also assume that the cost of sampling per fish is equal at both sampling occasions. If the costs of sampling per fish differ at each sampling occasion, then Robson and Regier (1964) and Seber (1982, p. 69) show that the optimal allocation of effort between the two sampling occasions is the solution to:

$$\frac{c_1 n_1}{c_2 n_2} = \frac{N - n_2}{N - n_1} \quad (2)$$

where c_1 and c_2 are the costs of sampling per fish at the two occasions. In many cases, the sample sizes are negligible compared to the abundance so the right hand side of Equation 2 is 1, and the optimal allocation reduces to spending equal amounts of money at the two sampling occasions.

Simple Petersen Sample Size Determination	
Use this spreadsheet to help plan experiments.	
Enter values for the approximate population size, and the amount of marks (entries in yellow) you can apply at time 1 and recover at time 2.	
Modify these values until the relative standard error (green) and the size of the confidence interval (tan) are acceptable.	
As a rough rule of thumb,	
<ul style="list-style-type: none"> - preliminary surveys should have 95% confidence intervals that are +/- 50% - management surveys should have 95% confidence intervals that are +/- 25% - scientific surveys should have 95% confidence intervals that are +/- 10% 	
0.95	1- α =confidence level
50,000	Population size
1,000	n ₁
3,000	n ₂
60	Expected number of marks recovered E[m ₂]
6,091	Estimated standard error
12%	Relative se
11,938	95% confidence interval is +/- this amount
24%	Relative size of the 95% confidence interval

Figure 2: Illustration of sample size determination worksheet for population of $N = 5,000$ with $n_1 = 1,000$ and $n_2 = 3,000$ fish captured at the two sample occasions.

Sample size guidelines 95% ci to be within 10% of correct result

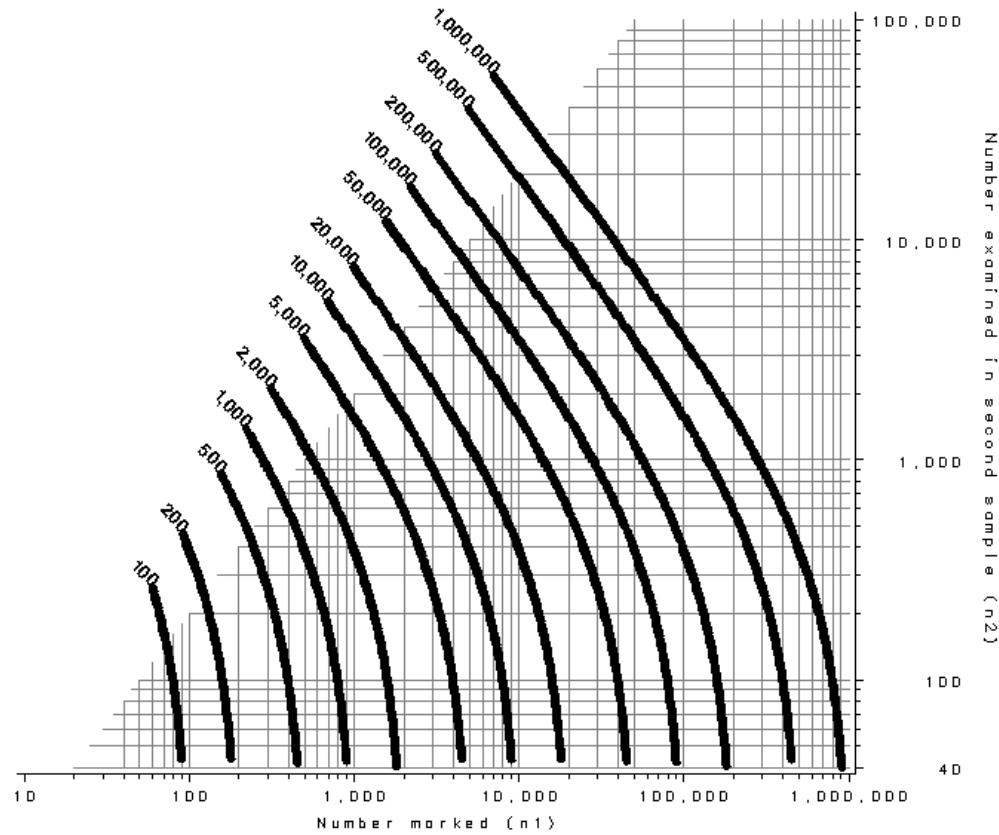


Figure 3: Sample size nomogram for 10% relative 95% confidence intervals suitable for scientific work.

Sample size guidelines 95% ci to be within 25% of correct result

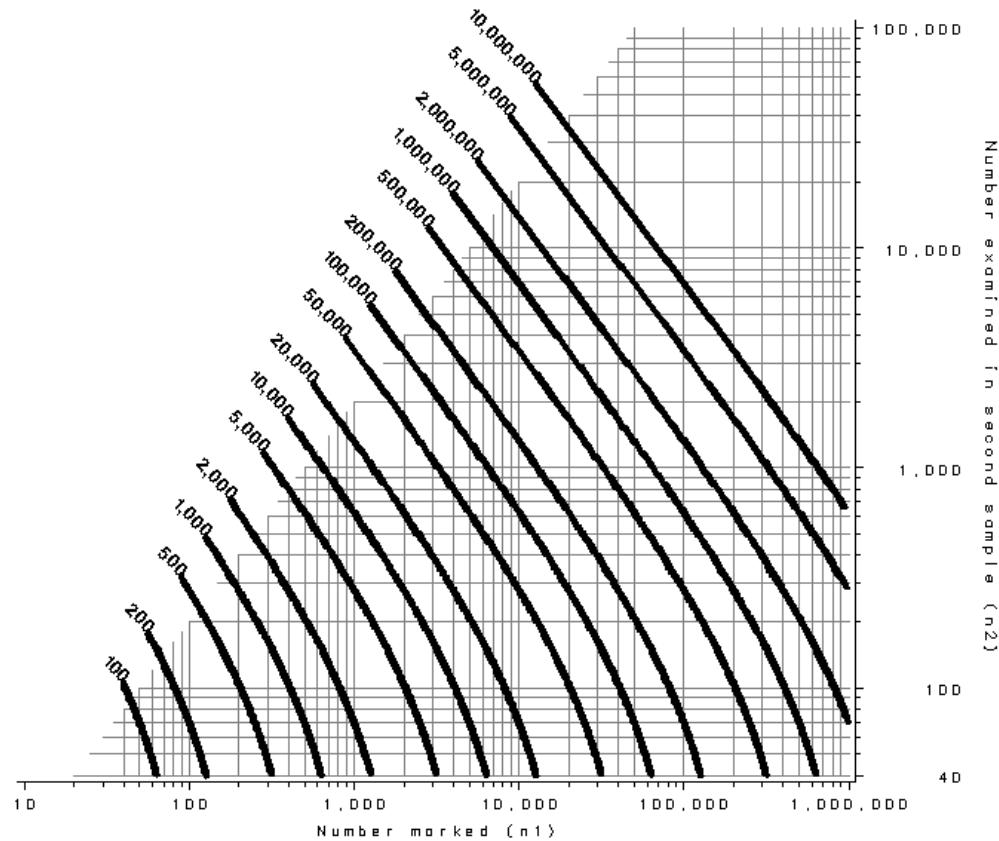


Figure 4: Sample size nomogram for 25% relative 95% confidence intervals suitable for management work.

Sample size guidelines 95% ci to be within 50% of correct result

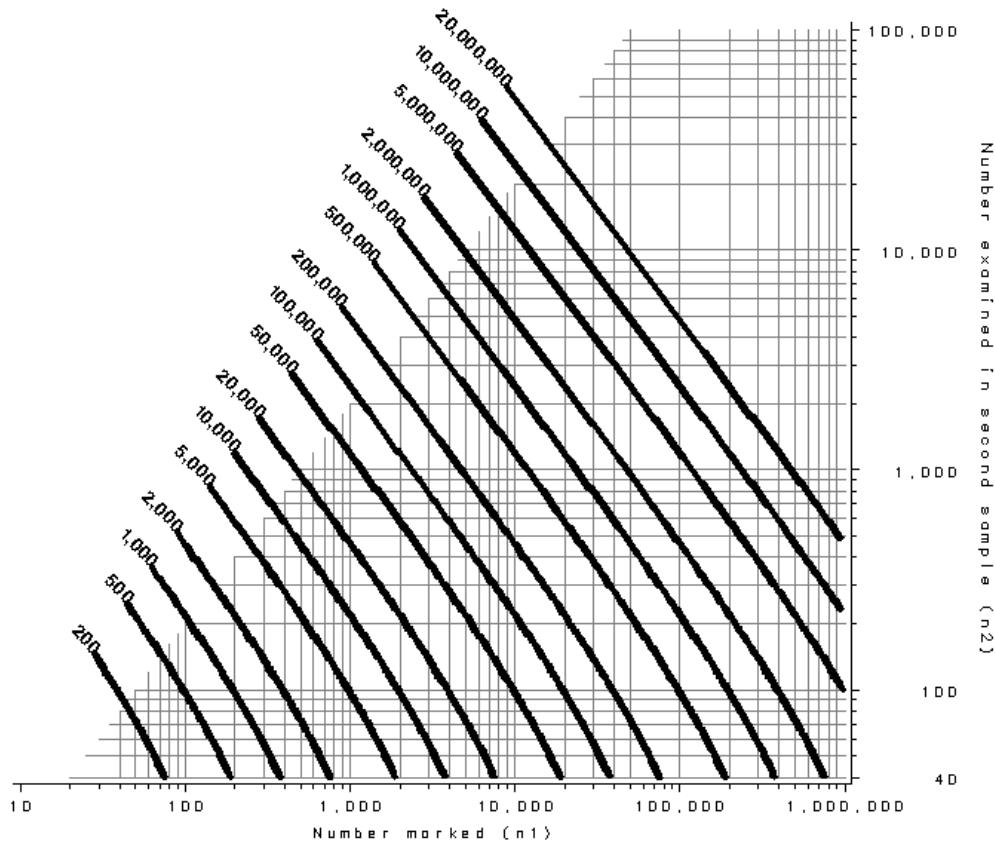


Figure 5: Sample size nomogram for 50% relative 95% confidence intervals suitable for preliminary surveys.

5 Assumptions and effects of violations

5.1 Introduction

The Petersen estimator makes a variety of assumptions and virtually no real study satisfies all of the assumptions. Consequently, a study of the effect of violations of assumptions upon the performance of the estimator is helpful in deciding how much credence should be placed upon the results.

In the sections that follow, the effect of violations of assumptions will be studied by substituting in the expected value of the statistics under the assumption violation. While this yields only an approximate answer to the effect of the violation, the results are sufficiently accurate to be useful in understanding.

In the workbook that accompanies this monograph, a specialized spreadsheet has been set up where the various effects (both individually and combined) can be studied without worrying about the hand computations (Section 5.7).

This is useful in order to see if violations of assumptions will lead to unacceptable bias. As suggested by Cochran (1977, p. 12-15), bias can typically be ignored if it is less than 10% of the standard error of the estimator.

There have been many papers that demonstrate how to modify the Petersen study to check for assumption violations and adjust the estimator – some of these are studied in future sections.

An ad-hoc adjustment can be made to the basic Petersen estimator if information is available about an assumption violation, e.g., an estimate of tag retention with accompanying SE. This is implemented in the *LP_est_adjust()* function as illustrated in Section 5.8.

5.2 Non-closure

The Petersen estimator assumes that the population of interest is closed. This means that no animals leave the population between the two sampling occasions through death or emigration, and no animals enter the population between the two sampling occasions through immigration or birth.

5.2.1 Immediate handling mortality

For many species, the act of catching and marking the fish traumatic and some fish suffer immediate mortality. In this case, the number of deaths is known. The study population should be reduced by these known mortalities and the resulting estimator should be conditional upon the actual number of fish released with tags rather than those captured.

The effect of marking upon subsequent mortality will be considered in Section 5.3.

5.2.2 Natural mortality or emigration

Suppose that natural mortality or emigration takes place between the two sampling occasions. These two types of non-closure are indistinguishable and shall be referred to as **mortality** in the remainder of this section.

What is the effect of this natural mortality if both the marked and unmarked fish have the same average survival probability between the two sampling occasions?

Let ϕ represent the average survival probability between the two sampling occasions. Then

- $E[n_1] \approx Np_1$
- $E[n_2] \approx [Np_1\phi + N(1-p_1)\phi] p_2$
- $E[m_2] \approx n_1\phi p_2 = Np_1\phi p_2$

and

$$E[\widehat{N}] \approx \frac{E[n_1] E[n_2]}{E[m_2]} = \frac{Np_1 \times N\phi p_2}{Np_1\phi p_2} = N$$

i.e., the estimator remains essentially unbiased for the abundance at the time of the first sampling occasion.

In many cases, the mortality probability varies among groups (strata) of the population, e.g., different mortalities across different age groups. If the marked fish can be divided into groups (strata), a simple $2 \times K$ contingency table can be constructed as shown in Table 4.

Table 4: Contingency table to test if mortalities are similar across different groups

	<i>A</i>	<i>B</i>	...	<i>K</i>	Total
Number released	n_{1A}	n_{1B}	...	n_{1K}	n_1
Recaptured	m_{2A}	m_{2B}	...	m_{2K}	m_2
Not recaptured	$n_{1A} - m_{2A}$	$n_{1B} - m_{2B}$...	$n_{1K} - m_{1K}$	$n_1 - m_2$

The usual χ^2 statistic is computed and is used to test if the product of survival and subsequent recapture ($\phi_x p_{2x}$) are equal across all groups (strata). If the hypothesis is rejected, then this MAY be evidence that the mortality probabilities are different in the subgroups – however it could also be evidence that the recapture probabilities are also unequal in the second sample across groups.

5.2.3 Immigration or births

Closure can also be violated by the addition of new animals, either by immigration from outside the study area or natural “births” (e.g., fish recruiting through growth). As both cases are indistinguishable, the term immigration will be used to refer to any increase in the population between the two sampling occasions.

Let λ be the rate of population increase between the two sampling occasions, i.e., an average of λ new fish enter the population per member of the original population. Then

- $E[n_1] \approx Np_1$
- $E[n_2] \approx N\lambda p_2$
- $E[m_2] \approx n_1\phi p_2 = Np_1\phi p_2$

and

$$E[\widehat{N}] \approx \frac{E[n_1]E[n_2]}{E[m_2]} = \frac{Np_1 \times N\lambda p_2}{Np_1 p_2} = N\lambda = N_2$$

i.e., the Petersen estimator is approximately unbiased for the abundance at the second sampling occasion.

In many cases, recruitment varies across different groups of the population, e.g., smaller ages may tend to recruit as they get older between the two sample occasions.

If the fish recovered in the second sample can be divided into groups (strata), a simple $2 \times K$ contingency table can be constructed as shown in Table 5.

Table 5: Contingency table to test for presence of recruitment among groups

	A	B	...	K	Total
Number captured	n_{2A}	n_{2B}	...	n_{2K}	n_2
Marked	m_{2A}	m_{2B}	...	m_{2K}	m_2
Not marked	$n_{2A} - m_{2A}$	$n_{2B} - m_{2B}$...	$n_{2K} - m_{1K}$	$n_2 - m_2$

The usual χ^2 statistic is computed and is used to test if the marked fraction are equals across all groups (strata). If the hypothesis is rejected , then this MAY be evidence of differential recruitment into the groups.

Seber (1982) has a non-parametric test for recruitment using length

5.2.4 Both immigration and mortality

Of course, both immigration and mortality can be occurring simultaneously.

As before, let ϕ represent the average survival probability between the two sampling occasions, and let λ represent the net recruitment per individual alive at the start of the study before any mortality occurs. Then

- $E[n_1] \approx Np_1$
- $E[n_2] \approx [N(\lambda - 1) + N\phi] p_2$
- $E[m_2] \approx n_1\phi p_2 = Np_1\phi p_2$

and

$$E[\widehat{N}] \approx \frac{E[n_1]E[n_2]}{E[m_2]} = \frac{Np_1 \times [N(\lambda - 1) + N\phi]p_2}{Np_1\phi p_2} = \frac{N[\lambda - 1 + \phi]}{\phi}$$

i.e., the estimator now estimates the total number of animals ever alive over the course of the study being a combination of the initial abundance and the net number of recruits..

5.3 Marking has no effect

The physical act of marking a fish can be very traumatic to individual fish. Marking effects usually take two forms (both of which may be present in a study) (a) a change in subsequent survival or (b) a change in subsequent catchability.

5.3.1 Marking affects survival

Both an acute effect (immediate mortality after release) and a chronic effect (no immediate mortality but increased mortality between the two sampling occasions compared to unmarked fish) can be handled in the same way.

Let ϕ represent the survival probability for unmarked fish between the two sampling occasions, and ϕ' represent the survival probability for marked fish between the two sampling occasions.

Then

- $E[n_1] \approx Np_1$
- $E[n_2] = [Np_1\phi' + N(1 - p_1)\phi] p_2$
- $E[m_2] = Np_1\phi' p_2$

and

$$E[\widehat{N}] \approx \frac{E[n_1]E[n_2]}{E[m_2]} = \frac{Np_1 \times [Np_1\phi' + N(1-p_1)\phi]p_2}{Np_1\phi'p_2} = N \left[\frac{\phi}{\phi'} + p_1 \left(1 - \frac{\phi}{\phi'} \right) \right]$$

If the two survival probabilities are equal, then the ratios of survival probabilities are all equal to 1 and the estimator gives the number alive at the first sampling occasion as seen earlier. If the survival probability of marked fish is lower than the survival probability of unmarked fish, then the ratios are all greater than 1, and the Petersen estimator over estimates the number of fish. This is intuitive as increased mortality among the the marked fish leads to fewer marked fish being recaptured than expected and an inflation in the estimate.

In many studies p_1 is small, and so the approximate relative bias is a function of the ratio of the two survival probabilities.

5.3.2 Marking affect catchability

The marking effect could affect subsequent catchability of the fish. This is often seen in small mammal studies where animals become trap happy or trap shy.

Let p_2 represent the catchability at the second sampling occasion for unmarked fish, and p_2^* represent the catchability at the second sampling occasion for marked fish. Then

- $E[n_1] \approx Np_1$
- $E[n_2] = Np_1p_2' + N(1-p_1)p_2$
- $E[m_2] = Np_1p_2'$

and

$$\begin{aligned} E[\widehat{N}] &\approx \frac{E[n_1]E[n_2]}{E[m_2]} = \frac{Np_1 \times [Np_1p_2' + N(1-p_1)p_2]}{Np_1p_2'} \\ &= N \left[\frac{p_2}{p_2'} + p_1 \left(1 - \frac{p_2}{p_2'} \right) \right] \end{aligned}$$

If the two catchabilities are equal, the ratios are all one in the last expression, and the estimator is approximately unbiased. If fish become trap shy, then the ratios are both greater than unity, and the estimator again overestimates the abundance. Intuitively, trap shyness reduces the observed number of marks below what is expected and inflates the estimated abundance.

Again, if the capture probability at the first sampling event (p_1) is small, the relative bias can be approximated by the ratio of the subsequent catchabilities.

5.4 Tag loss

Tags can be lost for a variety of reasons, e.g. breakage or tearing from the fish.

Let ρ represent the probability of **retaining** a tag between the two sampling occasions. Then

- $E[n_1] \approx Np_1$
- $E[n_2] \approx Np_2$
- $E[m_2] = Np_1\rho p_2$

and

$$E[\widehat{N}] \approx \frac{E[n_1] E[n_2]}{E[m_2]} = \frac{Np_1 \times Np_2}{Np_1\rho p_2} = \frac{N}{\rho}$$

Intuitively, tag loss results in fewer tags being observed in the second sample than expected with a consequent positive bias in the estimated abundance. Using the rule of thumb from Cochran (1977), the bias resulting from tag loss can be “ignored” if

$$\text{tag loss probability} = 1 - \rho < \frac{0.1}{\sqrt{E[m_T]}}$$

where m_T is the number of *marks* actually recovered.

Tag loss can be detected by double tagging all or a fraction of the fish released (Section 10). This second tag can be a batch mark (e.g. a fin clip) or a second tag of the same or different tag material (e.g., a disc tag could be used for the second tag if the first tag was a spaghetti tag).

5.5 Tags not reported or overlooked

In some case, sampling in the second sample is done by anglers or volunteers. Tags can be overlooked. Generically, this problem is referred to a **non-reporting** of tags. The effects of non-reporting of tags is identical to that of tag-loss in Section 5.4 and so a detailed analysis will not be repeated here.

Tag non-reporting can be estimated and adjusted for in a number of ways

- A sub sample of the second sample can be reexamined for missed tag to estimate how many tags have been missed.
- Two classes of tags can be applied, one of which is expected to have a 100% reporting probability (e.g. high value reward tags).

5.5.1 Inspecting a sub-sample of “un>tagged” fish

Rajwani and Schwarz (1997) considered the situation where a sub-sample of fish that supposedly did not have tags is reinspected to count the number of missed tags. An “exact” analysis is found in the spreadsheet that ships with this package and is not detailed here.

Rajwani and Schwarz (1997) and the accompanying spreadsheet also derived an optimal allocation of effort between the initial and secondary search to minimize the uncertainty in \widehat{N} for a given total budget.

An example of an empirical adjustment for non-reporting is found in Section [5.8.2](#).

5.5.2 Reward tags

In some cases, returns of tags are from anglers or other citizens and not all tags from recovered fish may be reported. To estimate the reporting probability, a second type of tag is also used (the “reward” tag) that offers a (monetary) incentive for its return. The incentive should be large enough to ensure that 100% of reward tags from recaptured fish are returned.

This type of study is exactly equivalent to a tag loss study and similar methods can be used. An example of the analysis of a reward tag study is found in Section [10.7](#).

5.6 Homogeneity in catchability

Variable catchability is one of the major problems in capture-recapture methods. Heterogeneity in catchability has been divided into two categories – pure heterogeneity where catchability varies among fish but the relative catchability of individual fish among themselves does not change between sampling occasions, and variable heterogeneity where the catchability of individuals varying within sampling occasions and the relative catchability of individual may vary across sampling occasions.

5.6.1 Pure heterogeneity

In pure heterogeneity, fish vary in their catchability due to random chance or related to covariates such as sex or body length. However, the relative catchability of individuals compared to other individuals remains fixed over the course of the study. For example, males may be more catchable than females, and remain more catchable in both sampling occasions. Or nets may be used to select fish, and net selectivity is related to body length which doesn’t change much between sampling occasions.

To illustrate the effects of heterogeneity, suppose that there are two sub-populations with a fraction a in the first sub-population and $1 - a$ in the second sub-population. Let p_1 and p_2 represent the catchability of the first subpopulation at the two sampling occasions, and suppose

that the second sub-population has catchabilities bp_1 and bp_2 respectively. The constant b represents the differential catchability among the two sub-populations at the two sampling occasions.

Then

- $E[n_1] \approx Nap_1 + N(1-a)bp_1$
- $E[n_2] \approx Nap_2 + N(1-a)bp_2$
- $E[m_2] = Nap_1p_2 + N(1-a)bp_1bp_2$

and

$$\begin{aligned} E[\hat{N}] &\approx \frac{E[n_1]E[n_2]}{E[m_2]} \\ &= \frac{[Nap_1 + N(1-a)bp_1][Nap_2 + N(1-a)bp_2]}{Nap_1p_2 + N(1-a)bp_1bp_2} \\ &= N \frac{[a+b(1-a)]^2}{a+b^2(1-a)} < N \end{aligned}$$

In general, pure heterogeneity leads to a negative bias in the Petersen estimator. Intuitively, the more catchable fish are caught too often which leads to an increase in the observed number of marks compared to homogeneous catchable populations and a deflation of the estimator. Indeed, if certain fractions of the population are uncatchable (e.g., $b = 0$) then the population estimates will always EXCLUDE the uncatchable segment and estimates only Na . If the second sub-population has the same catchability as the first population $a = 1$, then there is no bias (as expected).

If the heterogeneity is related to a fixed categorical covariate (such as sex), then the bias can be removed by stratifying the population by categories and performing independent estimates for each category (see Section 6).

In many cases, heterogeneity is related to a continuous covariate such as body length and induced by net selectivity. This continuous covariate can be used to model the catchability. Alternatively, Chen and Lloyd (2000) developed a non-parametric method to account for this type of heterogeneity – see Section 7.3 for details.

5.6.2 Variable heterogeneity

5.6.2.1 General heterogeneity

Both the cases above (pure heterogeneity and changing heterogeneity) can be subsumed into a general expression for the bias introduced by heterogeneity.

This can be generalized to a continuous distribution of catchabilities, say as a function of body length. Junge (1963) and Seber (1982, p. 86) show that the relative bias ($RB = \frac{E[\hat{N}]-N}{N}$) can

be approximated by:

$$RB \approx -C(p_{1j}, p_{2j}) \times \frac{\sqrt{V(p_{1j})V(p_{2j})}}{E[p_{1j}p_{2j}]}$$

where $C(\cdot, \cdot)$ is the correlation of catchabilities between sampling occasions, $V(\cdot)$ are the variability of the catchabilities at each sampling occasion, all taken over the individuals (j).

If all animals are equally catchable at either sampling occasion (e.g., a random sample at either sampling occasion), then the p_j are constant, the correlation is zero because any random variable has a 0 correlation with a constant. There is no bias.

If the catchabilities vary among individuals but the catchability in the second sample does not depend upon the catchability in the first sample, the correlation is again zero and there is no bias. This is reason for recommending that different sampling methods be used at each sampling occasion.

If the heterogeneity is related to a fish covariate such as size and the same sampling methods are used in both sampling occasions, then a positive correlation exists between the catchabilities and a negative bias in the estimate occurs.

Trap shyness leads to a negative correlation between the two capture probabilities, and a positive bias as seen earlier.

Junge(1963) and Seber(1982, p. 86) also examined how extreme the bias could be in the special case where $p_{2j} = bp_{1j}$, i.e., pure heterogeneity among animals. In this case the correlation is 1, and the relative bias simplifies to:

$$RB \approx -\frac{V(p_{1j})}{E[p_{1j}^2]}$$

This can be used to approximate the extent of the bias introduced by pure heterogeneity.

Expand on this here more, e.g. assume a beta distribution for p1 with various SD.

5.7 Spreadsheet to investigate impact of assumption violations

It is rare that a simple violation of assumptions will be occurring. The effect of multiple violations can be investigated using the spreadsheet supplied with this monograph.

The spreadsheet is set up to accommodate up to 4 segments of the population (e.g., strata), and has columns for the various assumption violations. The baseline condition is set up using guessestimates for the abundance and catchability probability and, as expected, no bias is seen in the estimates (Figure 6):

Now it is straight forward to investigate differences in catchability catchability and mortality between marked and unmarked fish. Suppose that marked fish are slightly less catchable at

								Immigration
								Pop change
								in absence of
10								
11								
12	Parameters							
13								
14	Popsize	1,000,000						
15			Prob of	Prob of				
16			capture	capture				
17	Group	Fraction of population	at time 1 (unmarked)	at time 2 (marked)	Survival between t1 -> t2	Survival between t1 -> t2	mortality between t1 -> t2	Tag retention between t1 -> t2
18			p1	p2	p2'	φ	φ*	λ
19	Entire pop	1.00	.007	.017	.017	1.000	1.000	1.000
20						1.000	1.000	1.000
21						1.000	1.000	1.000
22						1.000	1.000	1.000
23	Average/t	1.00	.007	.017	.017			
24	std dev							
25	c.v.							
26								

(a) Parameter values

51	Estimate - approximate value of estimate and potential bias		
52			
53	N-hat	1,000,000	Approximate expected value of estimate
54	se	90,569	Approximate expected se of estimate
55	relative se	9.1%	Approximate relative se of estimate
56			
57	Bias vs init	0.0%	Approximate bias relative to initial population size
58	Bias vs fin.	0.0%	Approximate bias relative to final population size
59			
60	Cochran (1977, page 14) indicates that if the bias < se/10, then its effect can be ignored.		

(b) Estimated bias

Figure 6: Baseline conditions for assessing impact of assumption violations

	A	B	C	D	E	F	G	H	I	J	K	L	M
1	Effect of violations of assumptions upon the Petersen Estimator												
2													
3	Use this spreadsheet to investigate the effect of violation assumptions upon the bias of the Petersen Estimator.												
4													
5	Because heterogeneity of catchability require the population to be divided into groups, this spreadsheet uses up to 4 groups in your population. This is often sufficient to check the gross effects of violations in assumptions.												
6													
7	The comment fields for each parameter column, explain what the various options mean.												
8													
9	The comment fields for each parameter column, explain what the various options mean.												
10													
11	Parameters												
12	Immigration Pop change in absence of												
13	Tag												
14	Popsiz	1,000,000		Prob of	Prob of	Survival	Survival	mortality					
15				capture	capture	between	between	between					
16				at time 2	at time 2	t1 -> t2	t1 -> t2	t1 -> t2					
17	Group	Fraction of population	at time 1	(unmarked)	(marked)	(unmarked)	(marked)	(per member)					
18		p1	p2	p2'	φ	φ*		λ					
19	Entire Pop	1.00	.007	.017	.016	1.000	.950	1.000	.950				
20	-	.00	.000	.000	.000	1.000	1.000	1.000	1.000				
21	-	.00	.000	.000	.000	1.000	1.000	1.000	1.000				
22	-	.00	.000	.000	.000	1.000	1.000	1.000	1.000				
23	Average/total	1.00	.007	.017	.016								
24	std dev			.000	.000	.000							
25	c.v.		0%	0%	0%								
26													

(a) Parameter values

51	Estimate - approximate value of estimate and potential bias												
52													
53	N-hat	1,176,413		Approximate expected value of estimate									
54	se	115,817		Approximate expected se of estimate									
55	relative se	9.8%		Approximate relative se of estimate									
56													
57	Bias vs initial	17.6%		Approximate bias relative to initial population size									
58	Bias vs final	17.7%		Approximate bias relative to final population size									
59													
60	Cochran (1977, page 14) indicates that if the bias < se/10, then its effect can be ignored.												

(b) Estimated bias

Figure 7: Impact of violations of assumptions

the second sampling event ($p_2^{\text{marked}} = .016$; $p_2^{\text{unmarked}} = .017$) and that marked fish have a survival probability of 0.95. The resulting bias is around 17% (Figure 7):

If heterogeneity is related to a fixed attribute, such as sex, the parameter values now are entered into 2 (or more) rows of the table. For examples, suppose we have a population with a 50:50 sex ratio; females are less catchable at the first event, and more catchable at the second event. The approximate bias is around 20% (Figure 8):

	A	B	C	D	E	F	G	H	I	J	K	L	M
1	Effect of violations of assumptions upon the Petersen Estimator												
2													
3	Use this spreadsheet to investigate the effect of violation assumptions upon the bias of the Petersen Estimator.												
4													
5	Because heterogeneity of catchability requires the population to be divided into groups, this spreadsheet uses up to 4 groups in your population. This is often sufficient to check the gross effects of violations in assumptions.												
6													
7													
8													
9	The comment fields for each parameter column, explain what the various options mean.												
10													
11													
12	Parameters												
13													
14	Popsize	1,000,000											
15			Prob of	Prob of									
16			capture	capture									
17	Group	Fraction of population	at time 1	(unmarked)	(marked)								
18						Immigration							
19						Pop change							
20						in absence of							
21													
22													
23	Average/total	1.00	.020	.014	.014								
24	std dev		.013	.004	.004								
25	c.v.		64%	26%	26%								
26													

(a) Parameter values

51	Estimate - approximate value of estimate and potential bias												
52													
53	N-hat	1,199,317				Approximate expected value of estimate							
54	se	79,836				Approximate expected se of estimate							
55	relative se	6.7%				Approximate relative se of estimate							
56													
57	Bias vs initial	19.9%				Approximate bias relative to initial population size							
58	Bias vs final	19.9%				Approximate bias relative to final population size							
59													
60	Cochran (1977, page 14) indicates that if the bias < se/10, then its effect can be ignored.												

(b) Estimated bias

Figure 8: Impact of heterogeneity associated with sex.

Finally, if heterogeneity is associated with geographic or temporal movement, a simple example allowing for a combination of 2 tagging strata (N vs S) and 2 recapture strata (N vs S) can be examined. Here we divide the population into 4 categories corresponding to the 4 possible

pairs of movements, and specify suitable values for the catchabilities. The estimated bias is around 5% (Figure 9):

							Immigration	
							Pop change	
							in absence of	
10								
11								
12	Parameters							
13								
14	Popsiz	1,000,000	Prob of	Prob of	Survival	Survival	mortality	Tag
15			capture	capture	between	between	between	
16	Fraction of		at time 2	at time 2	t1 -> t2	t1 -> t2	t1 -> t2	
17	Group	population	at time 1 (unmarked)	(marked)	(unmarked)	(marked)	per member	t1 -> t2
18			p1	p2	p2'	ϕ	ϕ*	λ
19	N->N	.30	.016	.025	.025	1.000	1.000	1.000
20	N->S	.30	.016	.004	.004	1.000	1.000	1.000
21	S->N	.10	.025	.025	.025	1.000	1.000	1.000
22	S->S	.30	.025	.004	.004	1.000	1.000	1.000
23	Average/t	1.00	.020	.012	.012			
24	std dev		.004	.010	.010			
25	c.v.		22%	83%	83%			

(a) Parameter values

51	Estimate - approximate value of estimate and potential bias
52	
53	N-hat 1,048,943 Approximate expected value of estimate
54	se 67,859 Approximate expected se of estimate
55	relative se 6.5% Approximate relative se of estimate
56	
57	Bias vs init 4.9% Approximate bias relative to initial population size
58	Bias vs fin. 4.9% Approximate bias relative to final population size
59	
60	Cochran (1977, page 14) indicates that if the bias < se/10, then its effect can be ignored.
61	

(b) Estimated bias

Figure 9: Impact of heterogeneity associated with geographic stratification.

Generally speaking, two tagging x two recapture strata will be sufficient to determine the approximate size of any bias.

5.8 Empirical adjustments to existing estimates

While it is possible to obtain a corrected point estimate using the analytical analysis above, finding the SE of the adjust estimate is more complex, especially if several adjustment factors are involved.

Consequently, and empirical adjustment can be obtained using the *LP_est_adjust()* function that takes the estimates (and SE) of abundance and estimates of the adjustment factor (and SE) and simulates the impact of the adjustments. A log-normal distribution is assumed for the estimates of abundance, and a distribution for each individual adjustment factor is determined using the corresponding estimate and SE. Once the effects of the combined adjustments is

simulated, the distribution of the abundance estimates is back-transformed and the adjusted SE (and confidence intervals) is determined. This process is analogous to what happens in a Bayesian analysis.

5.8.1 Adjusting for tag loss

5.8.1.1 Adjusting Rodli estimates for tag loss

Suppose we wish to adjust the Rodli estimates for tag loss. Recall the estimated abundance was 338 (SE 25) fish.

Suppose that the empirical estimate of tag retention is 0.90 (SE .05). The adjusted estimate is found as:

```
set.seed(23432)
rodli.est.mt.adjust <- LP_est_adjust(
  rodli.est.mt$summary$N_hat, rodli.est.mt$summary$N_hat_SE,
  tag.retention.est=0.90, tag.retention.se=0.05,
)
```

A comparison of the original and adjusted estimates of the abundance are:

```
rodli.est.mt.adjust$summary
```

	N_hat_un	N_hat_un_SE	N_hat_adj	N_hat_adj_SE	N_hat_adj_LCL	N_hat_adj_UCL
1	338.4737	25.49646	305.7388	28.62206	250.8642	364.0012

5.8.1.2 Example from a double tagging study

In Section 10.4.2, a double tagging study was conducted, and estimates of abundance and tag retention probability were obtained. We will re-analyze this data, using only 1 tag and then do the empirical adjustment (assuming we know the tag retention probability),

First we get the double tag data and remove information from the second tag (it doesn't matter that some revised histories are duplicated).

```
data(data_sim_tagloss_twoD)

data_one_tag <- data_sim_tagloss_twoD
data_one_tag$old_cap_hist <- data_one_tag$cap_hist
data_one_tag$cap_hist <- paste0(substr(data_one_tag$cap_hist,1,1),
                                substr(data_one_tag$cap_hist,3,3))
```

This gives the “single tag” data:

```
cap_hist freq old_cap_hist
1      01   879      0010
2      10   225      1000
3      11    14      1010
4      10   666      1100
5      10    21      1101
6      11     7      1110
7      11   37      1111
```

We now fit the regular Petersen estimator and get the estimates that are now biased because of tag loss:

```
data_one_tag.fit <- Petersen::LP_fit(data_one_tag, p_model=~1)
data_one_tag.est <- Petersen::LP_est(data_one_tag.fit)
data_one_tag.est$summary
```

```
N_hat_f      N_hat_rn      N_hat N_hat_SE N_hat_conf_level N_hat_conf_method
1 ~1 (Intercept) 15675.21 1933.056          0.95           logN
N_hat_LCL N_hat_UCL p_model name_model cond.ll n parms nobs method
1 12309.6 19961.03 ~1      p: ~1 -1499.301      1 1849 CondLik
```

And we make the empirical adjustment based on the results from the tag loss study:

```
data_one_tag.est.adj <- Petersen::LP_est_adjust(
  data_one_tag.est$summary$N_hat,
  data_one_tag.est$summary$N_hat_SE,
  tag.retention.est=.64,
  tag.retention.se =.06)
```

giving:

```
data_one_tag.est.adj$summary
```

```
N_hat_un N_hat_un_SE N_hat_adj N_hat_adj_SE N_hat_adj_LCL N_hat_adj_UCL
1 15675.21    1933.056 10099.55    1562.258 7339.145 13413.14
```

The estimate is similar to that in Section 10.4.2, but of course the SE is larger because the information from the other tags is not longer available.

5.8.2 Overlooked tags/Non-reporting of tags

Rajwani and Schwarz (1997) considered the situation where a sub-sample of fish that supposedly did not have tags is reinspected to count the number of missed tags. The data on males is taken from Rajwani and Schwarz (1997)

As fish return to their spawning sites, $n_1 = 1510$ are captured using seine nets. A Petersen disk tag is attached and the fish is released. After spawning, the fish die and the carcasses are often washed onto the banks of the spawning area. Survey teams walk along the banks looking for carcasses. When a carcass is found, it is examined for a tag. After enumeration, all tags are cut from the carcasses, and those carcasses are removed from the study area by cutting them into two with a machete and returning them to the river. Untagged carcasses are left where found.

A total of $n_2 = 45595$ carcasses are examined and $m_2 = 279$ marks are observed. Later in the season, a second team examines some of those carcasses identified as being without tags to check for tags missed in the initial survey. A total of $n_3 = 8462$ carcasses are reexamined (subsampled) and $m_3 = 6$ new tags are found.

The data and capture histories are:

	cap_hist	freq
1	10	1231
2	11	279
3	01	45316

We start with the standard conditional-likelihood Petersen estimate:

```
data_overlook.fit <- Petersen::LP_fit(data_overlook, p_model=~..time)
data_overlook.est <- Petersen::LP_est(data_overlook.fit)
data_overlook.est$summary
```

	N_hat_f	N_hat_rn	N_hat_N	N_hat_SE	N_hat_conf_level	N_hat_conf_method
1	~1	(Intercept)	246768.4	13298.26	0.95	logN
			N_hat_LCL	N_hat_UCL	p_model	name_model
1	222033.5	274258.7	~..time	p: ~..time	-7393.831	2
						46826 CondLik

This estimate is biased upwards because of the non-reporting of tags. Rajwani and Schwarz (1997) give the analytical expressions for the estimate of the reporting probability:

$$\hat{\rho} = \frac{m_2}{m_2 + n_{01} \frac{m_3}{n_3}} = \frac{279}{279 + 45316 \frac{6}{8462}}$$

We estimate the uncertainty in the reporting probability empirically:

```

# estimate the reporting rate
rho = m2/(m2+ (n2-m2)*m3/n3)

rho.sim <- m2/(m2+ (n2-m2)*(rbeta(10000,m3,n3-m3)))
rho.se = sd(rho.sim)

```

This gives an estimated reporting probability of 0.897 (SE 0.037).

These values are used to adjust the previous estimate

```

data_overlook.est.adj <- Petersen::LP_est_adjust(
  data_overlook.est$summary$N_hat,
  data_overlook.est$summary$N_hat_SE
  ,
  tag.reporting.est=rho,
  tag.reporting.se =rho.se)
data_overlook.est.adj$summary

```

	N_hat_un	N_hat_un_SE	N_hat_adj	N_hat_adj_SE	N_hat_adj_LCL	N_hat_adj_UCL
1	246768.4	13298.26	221357.3	14977.76	192665.1	250574.5

These results are very similar to those in Rajwani and Schwarz (1997) of $\widehat{N}=221,284$ (SE 15,068) fish.

5.8.3 Adjusting tags available using Bayesian methods

- refer to the Nass river study

6 Accounting for heterogeneity I - fixed discrete strata

Heterogeneity is likely the most common reason for extensive bias in the Petersen estimator. Pure heterogeneity, i.e., some fish are always more catchable than other fish, leads to a negative bias in the estimated abundance. Heterogeneity that varies among fish and between the two sample times can lead to positive or negative bias depending upon the correlation of the heterogeneity over the animals between the two sample times.

A common way to correct for biases caused by heterogeneity is stratification where animals are separated into groups by a measured covariate. There are four types of covariates that are commonly measured in (roughly) increasing order of complexity of analysis:

- fixed, individual categorical covariates such as sex.
- fixed, individual continuous covariates such as length in short studies.
- changing, individual, categorical covariates such as location or timing of capture at each sampling event.
- changing, individual continuous covariates such as length in long studies.

This section demonstrates the analysis of fixed, individual covariates such as sex.

6.1 Fixed individual categorical covariates such as sex

Very often a simple fixed categorical covariate is measured on individual fish such as sex at both sampling occasions. This device can also be used with continuous covariates (such as length) that do not change much over the course of a season if the continuous covariate is broken into distinct classes (e.g., length classes) as shown in Section 6.3.

A key assumption for this section is that fish do not change categories between the two sampling intervals.

6.1.1 Test for equal marked fractions

As a first step, a simple contingency table test can be used to examine if there is statistical evidence of differential catchability either in the first or second samples. Recall that the estimated catchabilities are found as $\hat{p}_1 = \frac{m_2}{n_2}$ and $\hat{p}_2 = \frac{m_2}{n_1}$. So, if there are K classes, a contingency table to test the equality of catchability at the first sample is equivalent to the test of presence of recruitment among groups found in Table 5 and is a test of equal marked fractions across the strata.

The usual χ^2 statistic is computed and is used to test if the marked fraction (the catchability at time 1), are equal across all strata. If the hypothesis is rejected , then this MAY be evidence of differential catchability at time 1.

6.1.2 Test for equal catchability

Similarly, the contingency table found in Table 4 can be used to test if the recapture probabilities at the second sampling occasion are equal.

6.1.3 Stratified models

If there is evidence that catchabilities may differ among the strata at least one sampling occasion, the simplest strategy is to compute separate Petersen estimates for each stratum and then combine the estimates. For example, if the population is stratified by sex and no assumption is made about the sex ratio at each sampling occasion, nor about equality of catchability at the sampling occasions, application of the previous methods lead to two estimates of abundance, one for each sex, and their associated standard error.

The combined estimate of abundance is found as:

$$\widehat{N}_{combined} = \widehat{N}_f + \widehat{N}_m$$

and the *SE* of the combined estimator is found as:

$$se(\widehat{N}_{combined}) = \sqrt{se(\widehat{N}_f)^2 + se(\widehat{N}_m)^2}$$

The extension to more than two categories is straightforward.

However, this strategy could be inefficient, if the catchability differs among the strata only at a single sampling occasion. For example, the catchability of both sexes may be the same at sample time 1, but differ at sample time 2.

There are four possible general models that could be fit. There are also several possible models where only some of the strata have equal catchability at either/both of the sampling occasions. The general theory presented here will also accommodate these cases.

- **Complete homogeneity:** $p_{1A} = p_{1B} = \dots = p_{1K}$ and $p_{2A} = p_{2B} = \dots = p_{2K}$.
- **Homogeneity at time 1 only:** Only $p_{1A} = p_{1B} = \dots = p_{1K}$.
- **Homogeneity at time 2 only:** Only $p_{2A} = p_{2B} = \dots = p_{2K}$.
- **Complete heterogeneity:** No equality at either sampling occasion.

The Akaike Information Criterion (AIC; Akaike, 1973) paradigm is a preferred method of dealing with multiple models for the same data. Burnham and Anderson (2002) provides a detailed reference on the use of this methodology. In brief, this paradigm asserts that none of the models fit to data are the “truth” (do you really believe that all individual of the same sex have exactly the same catchability). Rather all models are approximation to reality and several models may approximate reality almost equally well. The AIC statistic is computed for each model and is a measure of fit and a penalty for the number of parameters used to fit the model. The AIC statistic can be used to derive *model weights* which are a measure of relative strength in explaining the data among the competing models. A weighted average of the estimates from the competing model can be computed and the *SE* of this estimate can incorporate both the individual precision from each of the model plus a measure of model uncertainty. For example, if the various models give vastly different estimates of the population abundance, then the model uncertainty about the population abundance will be large. Conversely, if the various

models agree to a great extent in their estimate, then the uncertainty in the estimate due to model choice is small.

In some cases, additional information can, and should be used to improve precision. For example, in many species, the sex ratio in the population is known from other surveys or is assumed to be 50:50. I am unaware of any previous work on incorporating knowledge of the stratum population ratios into the estimation framework.

6.2 Example: Northern Pike - simple stratification

In 2005, the Minnesota Department of Natural Resources conducted a tagging study to estimate the number of northern pike (*Esox lucius*) in Mille Lacs, Minnesota. Briefly, approximately 7,000 fish were sexed and tagged on their spawning grounds in the spring and a summer gillnet assessment captured about 1,000 fish. Complete details are in Bruesewitz and Reeves (2005).

Fish were double tagged, but an analysis using a tag-retention model showed that tag loss was negligible (Section 10.6).

Each fish has its own individual history. The sex and length (inches) of the fish at the time of capture are also recorded. The first few records are:

```
data(data_NorthernPike)
head(data_NorthernPike)
##   cap_hist length Sex freq
## 1      01    23.20  M    1
## 2      01    28.89  F    1
## 3      01    25.20  M    1
## 4      01    22.20  M    1
## 5      01    25.00  M    1
## 6      01    24.70  M    1
```

6.2.1 Summary statistics by sex

The summary statistics by sex are found in Table 6.

Table 6: Summary statistics for Northern Pike capture-recapture study

Sex	n1	n2	m2	P(recap-ture)	Marked fraction
F	4,045	613	89	0.022	0.145
M	2,777	527	68	0.024	0.129

Sex	n1	n2	m2	P(recap-ture)	Marked fraction
ALL	6,822	1,140	157	0.023	0.138

The recapture probabilities are similar across the two sexes as are the marked fractions. However, with a large sample size, small differences in capture probabilities may be detected.

The analysis will be done using the capture histories for individual fish, but you can construct the capture histories from the summary statistics as shown below:

```
cap_hist <- n1_n2_m2_to_cap_hist(n1=c(4045,2777),
                                    n2=c(613,527),
                                    m2=c(89,68),
                                    strata=c("F","M"),stratum_var="Sex")
cap_hist
```

	cap_hist	freq	Sex
1	10	3956	F
2	10	2709	M
3	01	524	F
4	01	459	M
5	11	89	F
6	11	68	M

6.2.2 Test for equal marked fractions by sex

The contingency table to test for equal catchability at sampling occasion 1 (indicated by the marked fraction seen at the second occasion) is computed using the *LP_test_equal_mf()* function to test for equal marked fractions:

```
nop.equal.mf.sex <- LP_test_equal_mf(data_NorthernPike, "Sex")
```

This gives the summary table in Table 7.

Table 7: Contingency table for testing equal marked fraction

status	Number of fish		Proportions	
	F	M	F	M
Not seen at t1	524	459	0.855	0.871
Recaptured	89	68	0.145	0.129

and the resulting chi-square test output is:

```
## 
## Pearson's Chi-squared test with Yates' continuity correction
## 
## data: tab
## X-squared = 0.4942, df = 1, p-value = 0.4821
```

The χ^2 test p -value for equal marked fraction is $p = 0.482$ and so there is no evidence of differential marked fractions.

6.2.3 Test for equal recapture probabilities by sex

The contingency table to test for equal recapture probability from fish tagged at sampling occasion 1 is computed using the `LP_test_equal_recap()` function:

```
nop.equal.recap.sex <- LP_test_equal_recap(data_NorthernPike, "Sex")
```

This gives the summary table in Table 8.

Table 8: Contingency table for testing equal recapture probability

status	Number of fish		Proportions	
	F	M	F	M
Never seen	3,956	2,709	0.978	0.976

status	Number of fish		Proportions	
	F	M	F	M
Recaptured	89	68	0.022	0.024

and the results of the χ^2 test are:

```
## 
## Pearson's Chi-squared test with Yates' continuity correction
## 
## data: tab
## X-squared = 0.34826, df = 1, p-value = 0.5551
```

The p -value for equal recapture probability is $p = 0.555$ and so there is no evidence of differential recapture probabilities between the two sexes.

6.2.4 Fitting multiple models by sex

While these two tests indicate no evidence of differential catchability at either sampling occasion, they do not necessarily indicate that the two sexes can be completely pooled when computing the Petersen estimates.

Several models can be fit to the NorthernPike data:

- Completely pooled Petersen. Capture-probabilities at each sampling event do not depend on *Sex*. This is equivalent to fitting a simple Petersen model ignoring sex.
- Completely stratified-Petersen. Capture-probabilities at each sampling event depend on *Sex*. This is equivalent to fitting two separate Petersen models, one for each sex.
- Pure heterogeneous models where the probability of capture varies between the two sexes, but is consistently the same across sampling events. This is equivalent to an additive model between sex and time.

These models were fit to the northern pike data:

```
data(data_NorthernPike)

# Fit the various models

nop.fit.time      <- Petersen::LP_fit(data_NorthernPike, p_model=~..time)
nop.fit.sex.time  <- Petersen::LP_fit(data_NorthernPike, p_model=~`-1+Sex`..time)
```

```

nop.fit.sex.p.time <- Petersen::LP_fit(data_NorthernPike, p_model=~Sex+..time)

# Fit models where the p(capture) is equal at t1 or t2 but not both.
# This is intermediate between the ...time and ...time:Sex models

nop.fit.eq.t1      <- Petersen::LP_fit(data_NorthernPike,
                                         p_model=~1+I(as.numeric(..time==1))+  

                                         I(as.numeric(..time==2)):Sex)

nop.fit.eq.t2      <- Petersen::LP_fit(data_NorthernPike,
                                         p_model=~1+I(as.numeric(..time==2))+  

                                         I(as.numeric(..time==1)):Sex)

```

Notice the formula for the *p_model* to force the capture-probabilities to be equal at the first or second sampling event and to vary by sex for the other sampling event. The *I()* notation in the model formula indicates that the interior expression is evaluated first before being used in the model. In this case, we create an indicator variable if the sampling event is the first or second event.

We can now rank these models in terms of AICc (Table 9):

```

# compare the various models

nop.sex.aictab <- LP_AICc(
  nop.fit.time,
  nop.fit.sex.time,
  nop.fit.sex.p.time,
  nop.fit.eq.t1,
  nop.fit.eq.t2)

```

Table 9: Comparison of models fit to the Northern Pike data

Model specification	Conditi- onal log- likeliho	# parms	# obs	AICc	Delta	AICcWt
p: ~1 + I(as.numeric(..time == 2)) + I(as.numeric(..time == 1)):Sex	-3,696.051	3	7,805	7,398.11	0.00	0.42

Model specification	Condi-tional log-likelihood	# parms	# obs	AICc	Delta	AICcWt
p: ~-1 + I(as.numeric(..time == 1)) + I(as.numeric(..time == 2)):Sex	-3,696.139	3	7,805	7,398.28	0.18	0.38
p: ~-1 + Sex:.time	-3,695.826	4	7,805	7,399.66	1.55	0.19
p: ~..time	-3,702.341	2	7,805	7,408.68	10.58	0.00
p: ~Sex + ..time	-3,702.228	3	7,805	7,410.46	12.35	0.00

The AIC weight indicate high support for either model where the capture probabilities are equal either at the first or the second but not both sampling occasions. The model that involves complete pooling over both sexes (pooled Petersen) is given a very low weight (less than 1%). The model that has completely separate estimates for males and females is given a lower weight (only about 19%) as well.

We can now extract the estimates of the overall abundance from the models and obtain the model averaged values as shown in Table 10:

```
# extract the estimates of the overall abundance

nop.sex.ma.N_hat_all<- LP_modavg(
  nop.fit.time,
  nop.fit.sex.time,
  nop.fit.sex.p.time,
  nop.fit.eq.t1,
  nop.fit.eq.t2, N_hat=~1)
```

Table 10: Model averaged estimates of overall abundance for Northern Pike

N_hat_f	N_hat_rn	Modnames	AICcWt	Estimate	SE
~1	(Intercept)	p: ~-1 + I(as.numeric(..time == 2)) + I(as.numeric(..time == 1)):Sex	0.42	49,536	3,629
~1	(Intercept)	p: ~-1 + I(as.numeric(..time == 1)) + I(as.numeric(..time == 2)):Sex	0.38	49,536	3,629
~1	(Intercept)	p: ~-1 + Sex:.time	0.19	49,382	3,616

N_hat_f	N_hat_rn	Modnames	AICcWt	Estimate	SE
~1	(Intercept)	p: ~..time	0.00	49,536	3,629
~1	(Intercept)	p: ~Sex + ..time	0.00	49,596	3,643
		Model averaged		49,506	3,627

Notice that the estimates (and SE) for three of the models are identical – the pooled Petersen estimator (4th model), and the two model where the capture-probabilities are the same at either sampling event. The latter two models are one of the cases where the pooled-Petersen is unbiased, i.e., the capture-probabilities are homogeneous at either of the sampling events.

If only the model where each sex is modeled independently was used, the estimated abundances would have been 49,382 (SE 3,616) fish. In the models where the effect of sex was ignored either completely or at either of the sampling occasions, the estimated abundance would have been 49,536 (SE 3,629) fish. In this case, this extra work in trying to model the catchabilities did not lead to estimates that were very different than this most general model.

So what is the advantage of the first two models vs. the pooled-Petersen. It is now possible to obtain estimates of the abundance of the two sub-populations. In the traditional pooled-Petersen estimator, it is not easy to get estimates of the sub-population, but this can also be done using the conditional likelihood approach followed by the Horvitz-Thompson estimator (Table 11).

```
# extract the estimates of the abundance for each sex
nop.sex.ma.N_hat_sex<- LP_modavg(
  nop.fit.time,
  nop.fit.sex.time,
  nop.fit.sex.p.time,
  nop.fit.eq.t1,
  nop.fit.eq.t2, N_hat=~-1+Sex)
```

Table 11: Model averaged estimates of abundance for each sex for Northern Pike

N_hat_f	N_hat_rn	Modnames	AICcWt	Estimate	SE
~-1 + Sex	SexF	p: ~-1 + I(as.numeric(..time == 2)) + I(as.numeric(..time == 1)):Sex	0.42	26,814	2,048
~-1 + Sex	SexF	p: ~-1 + I(as.numeric(..time == 1)) + I(as.numeric(..time == 2)):Sex	0.38	29,338	2,171
~-1 + Sex	SexF	p: ~-1 + Sex:..time	0.19	27,860	2,700

N_hat_f	N_hat_rn	Modnames	AICcWt	Estimate	SE
~-1 + Sex	SexF	p: ~..time	0.00	28,998	2,139
~-1 + Sex	SexF	p: ~Sex + ..time	0.00	29,828	2,859
		Model averaged		27,993	2,506
~-1 + Sex	SexM	p: ~-1 + I(as.numeric(..time == 2)) + I(as.numeric(..time == 1)):Sex	0.42	22,722	1,823
~-1 + Sex	SexM	p: ~-1 + I(as.numeric(..time == 1)) + I(as.numeric(..time == 2)):Sex	0.38	20,197	1,499
~-1 + Sex	SexM	p: ~-1 + Sex:..time	0.19	21,522	2,406
~-1 + Sex	SexM	p: ~..time	0.00	20,538	1,526
~-1 + Sex	SexM	p: ~Sex + ..time	0.00	19,768	2,135
		Model averaged		21,513	2,160

Notice how estimates of abundance are requested for each sex. Now the estimates of the sub-population abundances differ among the models even though the estimates overall population abundances are equal

6.3 Example: Northern pike stratified by length class

We return to the Northern Pike example. The length of the fish was also measured and the sampling events are close enough that the change in length between the two sampling events is negligible.

6.3.1 Distribution of length in captured fish

A histogram of the distribution of length in the handled fish is show in Figure 10.

It appears that female fish tend to be larger than male fish.

6.3.2 Summary statistics by length class

We start by classifying the length into length classes of 0:20 in, 20:25 in, 25:30 in, 30:35 in; and 35+ in. Summary statistics are presented in @bl-nop-lengthclass-sumstat

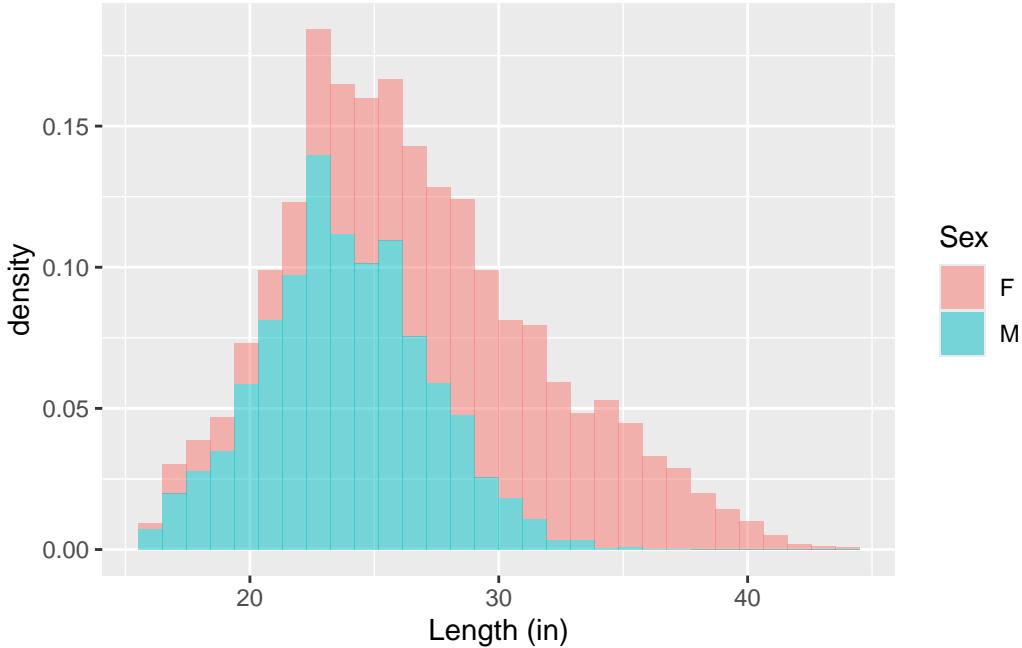


Figure 10: Histogram of length measurements for Northern Pike

Table 12: Summary statistics by length class

length.class	n1	n2	m2	P(recapture)	Marked fraction
00-20	569	22	1	0.002	0.045
20-25	2,238	371	41	0.018	0.111
25-30	2,120	537	86	0.041	0.160
30-35	1,254	173	25	0.020	0.145
35+	641	37	4	0.006	0.108
ALL	6,822	1,140	157	0.023	0.138

There appears to differences in recaptured probability by length class peaking around the 25-30 inch class, but the marked-fractions appear to be similar. Notice the small number of recaptures in the first and last length classes – these strata likely should be pooled with the other strata if a fully-stratified model is used to avoid the small sample biases.

6.3.3 Test for equal marked fractions by length class

The contingency table to test for equal catchability at sampling occasion 1 (indicated by the marked fraction seen at the second occasion) is constructed as illustrated in Table 13.

```
nop.equal.mf.length.class <- LP_test_equal_mf(data_NorthernPike,  
                                              "length.class")
```

Table 13: Contingency table for testing equal marked fraction

status	Number of fish					Proportions		
	00-20	20-25	25-30	30-35	35+	00-20	20-25	25-30
Not seen at t1	21	330	451	148	33	0.955	0.889	0.840
Recap- tured	1	41	86	25	4	0.045	0.111	0.160

and the resulting chi-square test output is:

```
##  
## Pearson's Chi-squared test  
##  
## data: tab  
## X-squared = 6.505, df = 4, p-value = 0.1645
```

The p -value for equal marked fraction is $p = 0.164$ and so there is no evidence of differential marked fractions among the length classes.

6.3.4 Test for equal recapture probabilities by length class

The contingency table to test for equal recapture probability from fish tagged at sampling occasion 1 is constructed as illustrated in Table 14.

```
nop.equal.recap.length.class <- LP_test_equal_recap(data_NorthernPike,  
                                              "length.class")
```

Table 14: Contingency table for testing equal recapture probability

status	Number of fish					Proportions		
	00-20	20-25	25-30	30-35	35+	00-20	20-25	25-30
Not recaptured	568	2,197	2,034	1,229	637	0.998	0.982	0.959
Recaptured	1	41	86	25	4	0.002	0.018	0.041

and the resulting chi-square test output is:

```
##
## Pearson's Chi-squared test
##
## data: tab
## X-squared = 51.225, df = 4, p-value = 2.003e-10
```

The p -value for equal recapture probability is $p < .001$ and so there appears to be good evidence of differential recapture probabilities among the length classes. However, some of the counts in some cells are quite small and so the p -value may be too small and a Fisher Exact test may be preferable.

```
##
## Fisher's Exact Test for Count Data with simulated p-value (based on
## 2000 replicates)
##
## data: tab
## p-value = 0.0004998
## alternative hypothesis: two.sided
```

The p -value is also small, so there is evidence of a difference in the recapture probabilities by length class.

6.3.5 Fitting multiple models by length class

We fit the suite of models similar to those when stratifying by sex as shown in Table 15.

```

# Fit the various models

nop.fit.time      <- Petersen::LP_fit(data_NorthernPike,
                                         p_model=~-1+..time)
nop.fit.length.class.time   <- Petersen::LP_fit(data_NorthernPike,
                                                 p_model=~-1+length.class:..time)
nop.fit.length.class.p.time <- Petersen::LP_fit(data_NorthernPike,
                                                 p_model=~length.class+..time)

# Fit models where the p(capture) is equal at t1 or t2 but not both.
# This is intermediate between the ...time and ...time:Sex models

nop.fit.eq.t1      <- Petersen::LP_fit(data_NorthernPike,
                                         p_model=~-1+I(as.numeric(..time==1))+
                                         I(as.numeric(..time==2)):length.class)

nop.fit.eq.t2      <- Petersen::LP_fit(data_NorthernPike,
                                         p_model=~-1+I(as.numeric(..time==2))+
                                         I(as.numeric(..time==1)):length.class)

```

We can now rank these models in terms of AICc (Table 15):

```

# compare the various models

nop.sex.aictab <- LP_AICc(
  nop.fit.time,
  nop.fit.length.class.time,
  nop.fit.length.class.p.time,
  nop.fit.eq.t1,
  nop.fit.eq.t2
)

```

Table 15: Comparison of models fit to the Northern Pike data stratifying by length class

Model specification	Conditi- onal log- likelihood	# parms	# obs	AICc	Delta	AICcW
p: ~-1 + I(as.numeric(..time == 1)) + I(as.numeric(..time == 2)):length.class	-3,596.138	6	7,805	7,204.29	0.00	0.62
p: ~-1 + length.class:..time	-3,592.615	10	7,805	7,205.26	0.97	0.38
p: ~-1 + I(as.numeric(..time == 2)) + I(as.numeric(..time == 1)):length.class	-3,621.220	6	7,805	7,254.45	50.16	0.00
p: ~length.class + ..time	-3,678.014	6	7,805	7,368.04	163.75	0.00
p: ~-1 + ..time	-3,702.341	2	7,805	7,408.68	204.40	0.00

The AIC weight indicate high support for either model where the capture probabilities are equal at the first sampling event (equal marked fraction) and secondarily, to a model with full stratification at both sampling events.

We can now extract the estimates of the overall abundance from the models and obtain the model averaged values as shown in Table 16:

```
# extract the estimates of the overall abundance

nop.length.class.ma.N_hat_all<- LP_modavg(
  nop.fit.time,
  nop.fit.length.class.time,
  nop.fit.length.class.p.time,
  nop.fit.eq.t1,
  nop.fit.eq.t2, N_hat=~1)
```

Table 16: Model averaged estimates of overall abundance for Northern Pike when stratifying by length class

N_hat_f	N_hat_rn	Modnames	AICcWt	Estimate	SE
~1	(Intercept)	p: ~-1 + I(as.numeric(..time == 1)) + I(as.numeric(..time == 2)):length.class	0.62	49,536	3,629
~1	(Intercept)	p: ~-1 + length.class:..time	0.38	60,614	13,039
~1	(Intercept)	p: ~-1 + I(as.numeric(..time == 2)) + I(as.numeric(..time == 1)):length.class	0.00	49,536	3,629
~1	(Intercept)	p: ~length.class + ..time	0.00	93,218	39,716
~1	(Intercept)	p: ~-1 + ..time	0.00	49,536	3,629
Model averaged				53,754	10,091

Notice that the estimates (and SE) for three of the models are identical – the pooled Petersen estimator, and the two model where the capture-probabilities are the same at either sampling event. The latter two models are one of the cases where the pooled-Petersen is unbiased, i.e., the capture-probabilities are homogeneous at either of the sampling events. However, there really is only support for the model with equal capture probability at the first sampling event.

Notice that the estimate for the fully stratified Petersen is quite large but with a very large standard error. This is likely an artefact of the small number of recaptures. It is left as an exercise for the reader to refit the models, pooling the first and last length classes with the second or second-to-last classes to avoid the small sample bias problem.

Finally, here are the estimates of abundance by length class using the the Horvitz-Thompson estimator (Table 17).

```
# extract the estimates of the abundance for each sex
nop.length.class.ma.N_hat_length.class<- LP_modavg(
  nop.fit.time,
  nop.fit.length.class.time,
  nop.fit.length.class.p.time,
  nop.fit.eq.t1,
  nop.fit.eq.t2, N_hat=~-1+length.class)
```

Table 17: Model averaged estimates of abundance for each length class for Northern Pike

N_hat_rn	Modnames	AICcWt	Estimate	SE
20	length.class00~-1 + I(as.numeric(..time == 1)) + I(as.numeric(..time == 2)):length.class	0.62	4,146	345
20	length.class00- p: ~-1 + length.class:..time	0.38	12,518	12,220
20	length.class00~-1 + I(as.numeric(..time == 2)) + I(as.numeric(..time == 1)):length.class	0.00	1,481	210
20	length.class00- p: ~length.class + ..time	0.00	39,449	39,004
20	length.class00- p: ~-1 + ..time	0.00	3,745	306
Model averaged			7,334	8,571
25	length.class00~-1 + I(as.numeric(..time == 1)) + I(as.numeric(..time == 2)):length.class	0.62	16,324	1,224
25	length.class20- p: ~-1 + length.class:..time	0.38	20,251	2,955
25	length.class00~-1 + I(as.numeric(..time == 2)) + I(as.numeric(..time == 1)):length.class	0.00	16,577	1,374
25	length.class20- p: ~length.class + ..time	0.00	20,014	2,849
25	length.class20- p: ~-1 + ..time	0.00	16,298	1,218
Model averaged			17,819	2,809
30	length.class05~-1 + I(as.numeric(..time == 1)) + I(as.numeric(..time == 2)):length.class	0.62	15,306	1,136
30	length.class25- p: ~-1 + length.class:..time	0.38	13,238	1,281
30	length.class05~-1 + I(as.numeric(..time == 2)) + I(as.numeric(..time == 1)):length.class	0.00	21,717	1,795
30	length.class25- p: ~length.class + ..time	0.00	10,614	952

N_hat_rn	Modnames	AICcWt	Estimate	SE
length.class25- 30	p: ~-1 + ..time	0.00	16,317	1,219
	Model averaged		14,519	1,560
length.class30- 35	p: ~-1 + I(as.numeric(..time == 1)) + I(as.numeric(..time == 2)):length.class	0.62	9,097	702
length.class30- 35	p: ~-1 + length.class:..time	0.38	8,678	1,589
length.class30- 35	p: ~-1 + I(as.numeric(..time == 2)) + I(as.numeric(..time == 1)):length.class	0.00	7,685	728
length.class30- 35	p: ~length.class + ..time	0.00	9,897	1,775
length.class30- 35	p: ~-1 + ..time	0.00	8,898	681
	Model averaged		8,937	1,144
length.class35+ 35	p: ~-1 + I(as.numeric(..time == 1)) + I(as.numeric(..time == 2)):length.class	0.62	4,662	383
length.class35+ 35	p: ~-1 + length.class:..time	0.38	5,929	2,791
length.class35+ 35	p: ~-1 + I(as.numeric(..time == 2)) + I(as.numeric(..time == 1)):length.class	0.00	2,075	271
length.class35+ length.class + ..time	p: ~length.class + ..time	0.00	13,244	6,369
length.class35+ 1	p: ~-1 + ..time	0.00	4,278	345
	Model averaged		5,145	1,853

Notice how estimates of abundance are requested for each length class. Now the estimates of the sub-population abundances differ among the models even though the estimates overall population abundances are equal

7 Accounting for heterogeneity II - continuous fixed covariate

In the previous sections, we showed how a continuous covariate could be broken into a set of discrete classes and a stratified model applied. In some cases, it may be of interest to use

a smooth function of the covariates (e.g., a quadratic curve, or a spline fit) to represent, for example, the catchability curve as a function of length.

This is relatively straight forward except for a few items that need to be addressed

- Standardizing the covariates to have a mean close to 0 and a standard deviation close to 1 improves numerical stability and convergence, especially when fitting a quadratic curve.
- Animals with a very low estimated catchability have a very large expansion factor when the Horvitz-Thompson estimator is formed. You may need to remove fish with very large expansion factors when estimating abundance.

7.1 Quadratic relationship between the probability of capture and length - Northern Pike

We again return to the Northern Pike example. We first standardize the *length* measurement (the *ls* variable in the revised data frame):

```
data(data_NorthernPike)

data_NorthernPike$ls <- scale(data_NorthernPike$length)
head(data_NorthernPike)
##   cap_hist length Sex freq      ls
## 1       01    23.20  M   1 -0.7146489
## 2       01    28.89  F   1  0.3710705
## 3       01    25.20  M   1 -0.3330252
## 4       01    22.20  M   1 -0.9054607
## 5       01    25.00  M   1 -0.3711876
## 6       01    24.70  M   1 -0.4284311
```

Now we fit several models and compare them to the previous models.

- a single quadratic curve (on the logit scale) with an additive shift between sampling events
- two separate quadratic curves (on the logit scale) for each sampling event

```
nop.fit.length.quad1 <- Petersen::LP_fit(data_NorthernPike,
                                             p_model=~..time + ls + I(ls^2))
nop.fit.length.quad2 <- Petersen::LP_fit(data_NorthernPike,
                                             p_model=~..time + ls + I(ls^2) +
                                             ..time:ls + ..time:I(ls^2))
```

We can compare models with the previous models using length classes using the usual AICc methods (Table 18).

```
# compare the various models

nop.sex.aictab <- LP_AICc(
  nop.fit.time,
  nop.fit.length.class.time,
  nop.fit.length.class.p.time,
  nop.fit.eq.t1,
  nop.fit.eq.t2,
  nop.fit.length.quad1,
  nop.fit.length.quad2)
```

Table 18: Comparison of models fit to the Northern Pike data including quadratic functions of length

Model specification	Conditional log-likelihood	# parms	# obs	AICc	Delta	AICcWt
p: ~..time + ls + I(ls^2) + ..time:ls + ..time:I(ls^2)	-3,575.331	6	7,805	7,162.67	0.00	1.00
p: ~-1 + I(as.numeric(..time == 1)) + I(as.numeric(..time == 2)):length.class	-3,596.138	6	7,805	7,204.29	41.61	0.00
p: ~-1 + length.class:..time	-3,592.615	10	7,805	7,205.26	42.59	0.00
p: ~-1 + I(as.numeric(..time == 2)) + I(as.numeric(..time == 1)):length.class	-3,621.220	6	7,805	7,254.45	91.78	0.00
p: ~length.class + ..time	-3,678.014	6	7,805	7,368.04	205.37	0.00
p: ~-1 + ..time	-3,702.341	2	7,805	7,408.68	246.01	0.00
p: ~..time + ls + I(ls * 2)	-3,702.290	4	7,805	7,412.59	249.91	0.00

The model with two separate quadratic fits is by far the best fitting model.

We obtain the estimate of abundance and look at the estimated relationship between the length and the probability of capture (Figure 11)

```

nop.est.length.quad2 <- Petersen::LP_est(nop.fit.length.quad2, N_hat=~1)

est.p <- nop.est.length.quad2$detail$data.expand
ggplot(data=est.p, aes(x=length, y=p, color=as.factor(..time)))+
  geom_point()+
  geom_line(size=.1)+ 
  scale_color_discrete(name="Sample\\nevent")+
  xlab("Length (in)")+
  ylab("Estimated p(capture)")

```

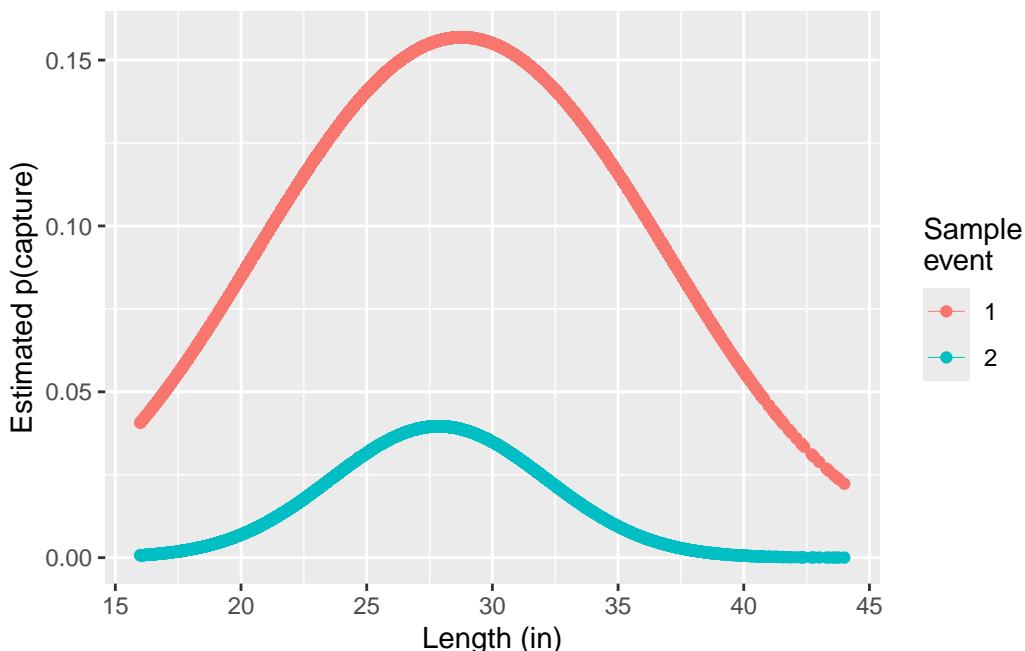


Figure 11: Estimated probability of capture from quadratic relationship with length

We see that the relationship at time 1 is quite peaked but not so much for sampling event 2 which is in accordance to the results from classifying length into discrete bins.

This model also shows a positive association between catchability at the two sampling events (Figure 12)

```

est.p <- nop.est.length.quad2$detail$data.expand
temp <- tidyr::pivot_wider(est.p,
                           id_cols=c(..index", "length"),
                           values_from="p",
                           names_from=".time",

```

```

  names_prefix="t")
ggplot(data=temp, aes(x=t1, y=t2, color=length))+  

  geom_point()+
  scale_color_continuous(name="Length")+
  xlab("Estimated p(capture) at sample event 1")+
  ylab("Estimated p(capture) at sample event 2")

```

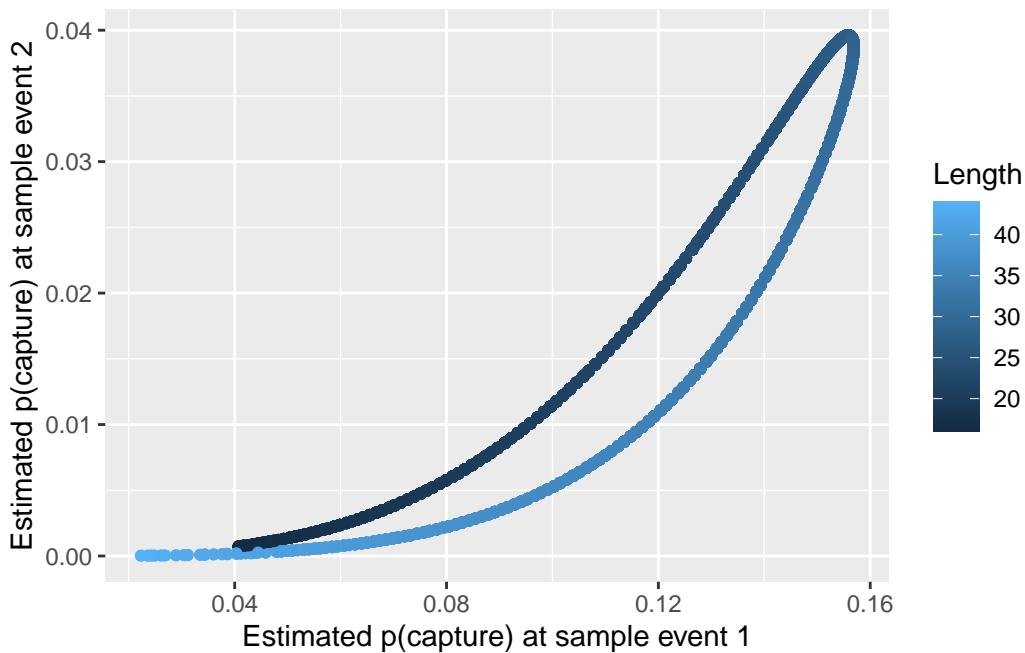


Figure 12: Estimated relationship between catchability at two sampling events

The odd shape to the relationship is an artefact of the quadratic curves where the top/bottom half of the “curve” correspond to the ascending/descending limbs of the quadratic curve seen in Figure 11.

Figure 12 implies a positive correlation between the two capture probabilities which implies that the pooled-Petersen estimator will have a negative bias as seen in the model averaged results (Table 20).

The estimated abundance from the best fitting model is:

The estimate appears on the large side with a large standard error compared to the other estimates seen previously. We suspect that some histories have a very small probability of capture and a large expansion factor as shown in Figure 13

Table 19: Estimated abundance of Northern Pike from quadratic relationship with length

```

nop.est.length.quad2$summary
##   N_hat_f      N_hat_rn      N_hat N_hat_SE N_hat_conf_level N_hat_conf_method
## 1 ~1 (Intercept) 59209.91 9365.551                      0.95                  logN
##   N_hat_LCL N_hat_UCL
## 1 43426.54 80729.73 ...time + ls + I(ls^2) + ...time:ls + ...time:I(ls^2)
##                                         name_model cond.ll n parms nobs
## 1 p: ...time + ls + I(ls^2) + ...time:ls + ...time:I(ls^2) -3575.331       6 7805
##   method
## 1 CondLik

```

```

est.ef <- nop.est.length.quad2$detail$data
ggplot(data=est.ef, aes(x=length, y=..EF))+
  geom_point()+
  geom_line()+
  xlab("Length (in)")+
  ylab("Estimated expansion factor")

```

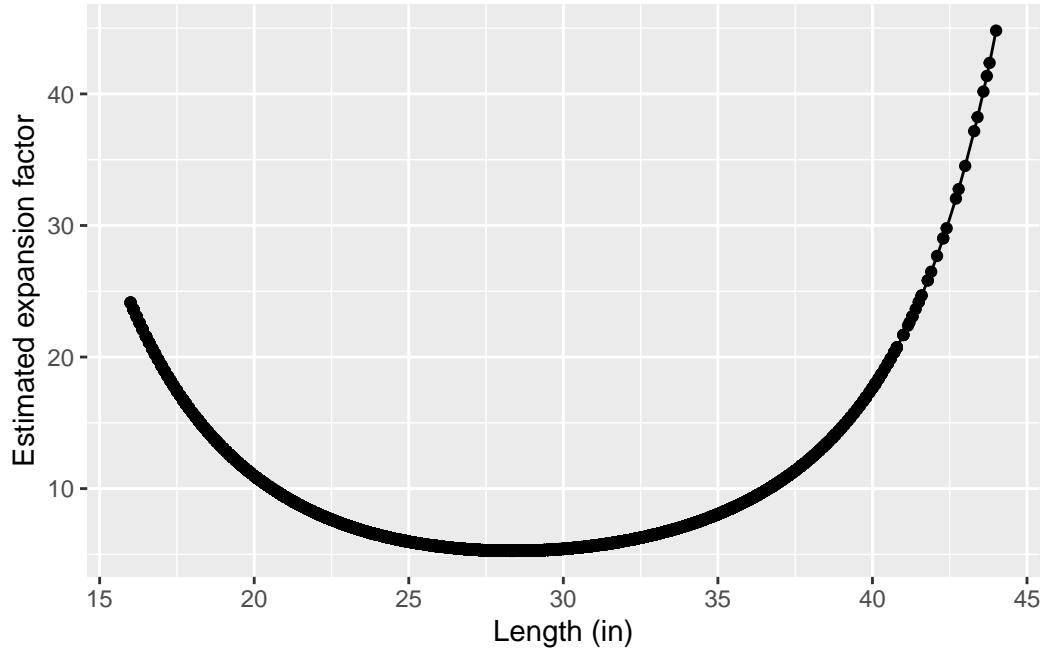


Figure 13: Estimated expansion factor by length from quadratic relationship with length

We can see that there are several large fish with expansion factors that appear to be much larger than the majority of fish. We can estimate the abundance truncating the expansion factor, say at 30:

```
nop.est.length.quad2.tef <- Petersen::LP_est(nop.fit.length.quad2,
                                              N_hat=~-1+I(as.numeric(..EF<30)))

plyr:::rbind.fill(nop.est.length.quad2      $summary,
                  nop.est.length.quad2.tef$summary)
##                                     N_hat_f          N_hat_rn      N_hat N_hat_SE
## 1                               ~1          (Intercept) 59209.91 9365.551
## 2 ~-1 + I(as.numeric(..EF < 30)) I(as.numeric(..EF < 30)) 58866.46 9005.852
##   N_hat_conf_level N_hat_conf_method N_hat_LCL N_hat_UCL
## 1           0.95          logN 43426.54 80729.73
## 2           0.95          logN 43615.86 79449.54
##                                         p_model
## 1 ~..time + ls + I(ls^2) + ..time:ls + ..time:I(ls^2)
## 2 ~..time + ls + I(ls^2) + ..time:ls + ..time:I(ls^2)
##                                     name_model cond.ll n parms nobs
## 1 p: ~..time + ls + I(ls^2) + ..time:ls + ..time:I(ls^2) -3575.331       6 7805
## 2 p: ~..time + ls + I(ls^2) + ..time:ls + ..time:I(ls^2) -3575.331       6 7805
##   method
## 1 CondLik
## 2 CondLik
```

Notice the use of the special variable ..EF in the model for N_{hat} . The estimate is reduced (as expected) but not by a great extent, so there is no real reason to adopt this second estimate.

Finally, we can again do the model averaging (Table 20):

```
# extract the estimates of the abundance
nop.length.class.ma.N_hat_length.class<- LP_modavg(
  nop.fit.time,
  nop.fit.length.class.time,
  nop.fit.length.class.p.time,
  nop.fit.eq.t1,
  nop.fit.eq.t2,
  nop.fit.length.quad1,
  nop.fit.length.quad2,    N_hat=~1)
```

Table 20: Model averaged estimates of abundance including the quadratic fit to length

Modnames	AICcWt	Estimate	SE
p: ~..time + ls + I(ls ^2) + ..time:ls + ..time:I(ls ^2)	1.00	59,210	9,366
p: ~-1 + I(as.numeric(..time == 1)) + I(as.numeric(..time == 2)):length.class	0.00	49,536	3,629
p: ~-1 + length.class:..time	0.00	60,614	13,039
p: ~-1 + I(as.numeric(..time == 2)) + I(as.numeric(..time == 1)):length.class	0.00	49,536	3,629
p: ~length.class + ..time	0.00	93,218	39,716
p: ~-1 + ..time	0.00	49,536	3,629
p: ~..time + ls + I(ls * 2)	0.00	49,563	3,635
Model averaged		59,210	9,366

The quadratic model is so much a better fit that it overwhelms the other models and the model averaged estimate is basically the first model. Notice that the model averaged confidence interval is larger than the original model because the variation in estimates among the models has also been taken into account.

It would be interesting to fit quadratic on $\log(\text{length})$ to give a skewed catchability curve.

7.2 Spline relationship between the probability of capture and length - Northern Pike

The quadratic curve fairly ridged in its shape. An alternate way to fit a curve to the catchabilities is through the use of splines.

Give a tutorial on splines here and how to choose the appropriate values for df etc - use AIC to select

We continue with the Northern Pike example and fit a model with a separate spline curve at each sample event.

```
nop.fit.length.sp2.df4<- Petersen::LP_fit(data_NorthernPike,  
                                         p_model=~..time+bs(ls, df=4) +  
                                         ..time:bs(ls, df=4))
```

```

nop.fit.length.sp2.df5<- Petersen::LP_fit(data_NorthernPike,
                                             p_model=~..time+bs(ls, df=5) +
                                             ..time:bs(ls, df=5))

```

We can compare models using the usual AICc methods (Table 21).

```

# compare the various models

nop.sex.aictab <- LP_AICc(
  nop.fit.time,
  nop.fit.length.class.time,
  nop.fit.length.class.p.time,
  nop.fit.eq.t1,
  nop.fit.eq.t2,
  nop.fit.length.quad1,
  nop.fit.length.quad2,
  nop.fit.length.sp2.df4, nop.fit.length.sp2.df5)

```

Table 21: Comparison of models fit to the Northern Pike data including spline functions of length

Model specification	Conditional log-likelihood	# parms	# obs	AICc	Delta	AICcWt
p: ~..time + bs(ls, df = 4) + ..time:bs(ls, df = 4)	-3,565.507	10	7,805	7,151.04	0.00	0.85
p: ~..time + bs(ls, df = 5) + ..time:bs(ls, df = 5)	-3,565.290	12	7,805	7,154.62	3.58	0.14
p: ~..time + ls + I(ls ^2) + ..time:ls + ..time:I(ls ^2)	-3,575.331	6	7,805	7,162.67	11.63	0.00
p: ~-1 + I(as.numeric(..time == 1)) + I(as.numeric(..time == 2)):length.class	-3,596.138	6	7,805	7,204.29	53.24	0.00
p: ~-1 + length.class:..time	-3,592.615	10	7,805	7,205.26	54.22	0.00

Model specification	Conditi- onal log- likelihood	# parms	# obs	AICc	Delta	AICcWt
p: ~-1 + I(as.numeric(..time == 2)) + I(as.numeric(..time == 1)):length.class	-3,621.220	6	7,805	7,254.45	103.41	0.00
p: ~length.class + ..time	-3,678.014	6	7,805	7,368.04	217.00	0.00
p: ~-1 + ..time	-3,702.341	2	7,805	7,408.68	257.64	0.00
p: ~..time + ls + I(ls * 2)	-3,702.290	4	7,805	7,412.59	261.54	0.00

The model with two separate splines and 4 df is again the best model in the set. We obtain the estimate of abundance and look at the estimated relationship between the length and the probability of capture (Figure 14)

```
nop.est.length.sp2.df4 <- Petersen::LP_est(nop.fit.length.sp2.df4, N_hat=~1)

est.p <- nop.est.length.sp2.df4$detail$data.expand
ggplot(data=est.p, aes(x=length, y=p, color=as.factor(..time)))+
  geom_point()+
  geom_line(size=.1)+
  scale_color_discrete(name="Sample\nevent")+
  xlab("Length (in)")+
  ylab("Estimated p(capture)")
```

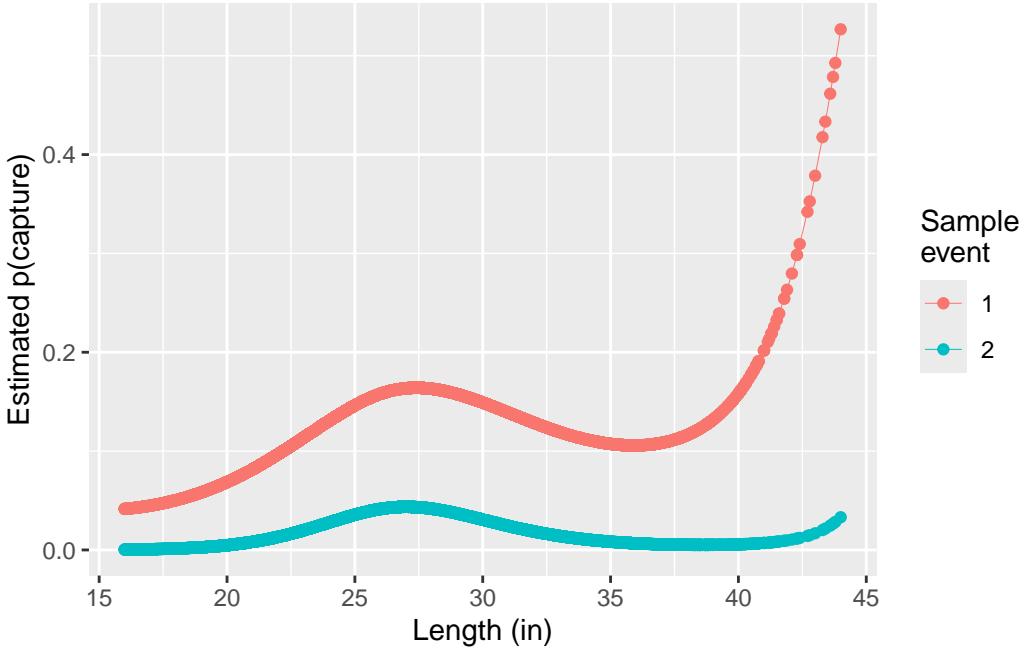


Figure 14: Estimated probability of capture from spline fit with length

We see that the relationship at time 1 is quite peaked but not so much for sampling event 2 which is in accordance to the results from classifying length into discrete bins. This spline model seems to indicate a very high catchability of very large fish at the first sampling event, something that was missed when the simple quadratic curve was used.

This model also shows a positive association between catchability at the two sampling events (Figure 15)

```
est.p <- nlp.est.length.sp2.df4$detail$data.expand
temp <- tidyr::pivot_wider(est.p,
                           id_cols=c(..index", "length"),
                           values_from="p",
                           names_from=".time",
                           names_prefix="t")
ggplot(data=temp, aes(x=t1, y=t2, color=length))+
  geom_point()+
  scale_color_continuous(name="Length")+
  xlab("Estimated p(capture) at sample event 1")+
  ylab("Estimated p(capture) at sample event 2")
```

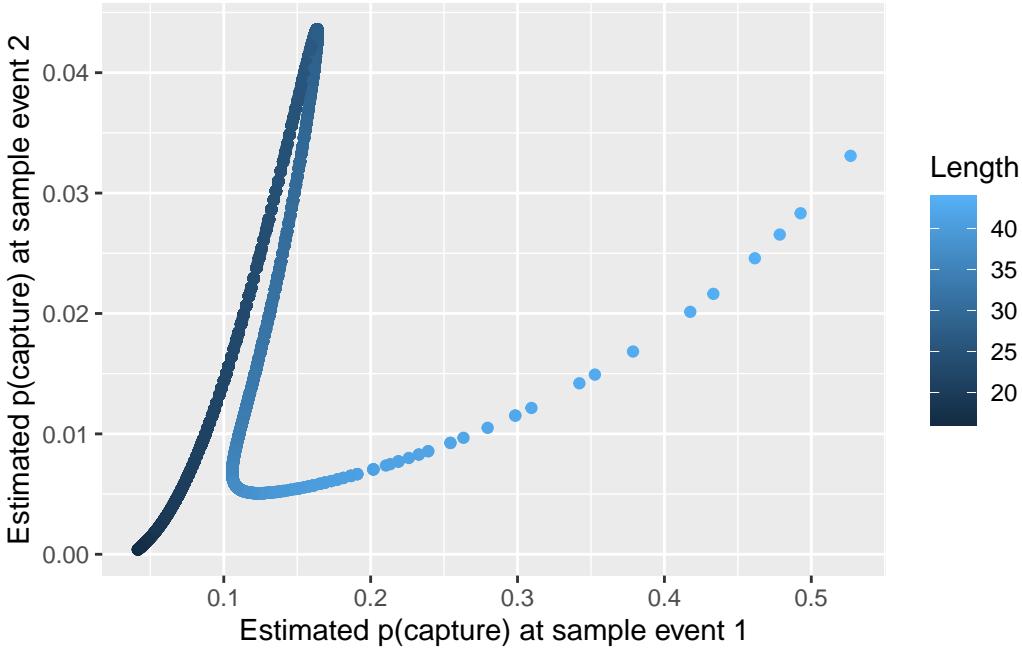


Figure 15: Estimated relationship between catchability at two sampling events from spline fit

The odd shape to the relationship is an artefact of two spline curves and the differing relationship between the ascending/descending arms of the spline. The above fit implies a positive correlation between the two capture probabilities which implies that the pooled-Petersen estimator will have a negative bias as seen in the model averaged results (Table 22).

```
nop.est.length.sp2.df4$summary
##   N_hat_f      N_hat_rn      N_hat N_hat_SE N_hat_conf_level N_hat_conf_method
## 1      ~1 (Intercept) 60485.41 12635.26                      0.95          logN
##   N_hat_LCL N_hat_UCL                               p_model
## 1 40163.97 91088.72 ~..time + bs(ls, df = 4) + ..time:bs(ls, df = 4)
##                                         name_model cond.ll n parms nobs
## 1 p: ~..time + bs(ls, df = 4) + ..time:bs(ls, df = 4) -3565.507      10 7805
##   method
## 1 CondLik
```

Finally, we can again do the model averaging (Table 22):

```
# extract the estimates of the abundance
nop.length.class.ma.N_hat_length.class<- LP_modavg(
  nop.fit.time,
  nop.fit.length.class.time,
```

```

nop.fit.length.class.p.time,
nop.fit.eq.t1,
nop.fit.eq.t2,
nop.fit.length.quad1,
nop.fit.length.quad2,
nop.fit.length.sp2.df4, nop.fit.length.sp2.df5, N_hat=~1)

```

Table 22: Model averaged estimates of abundance including the spline fit to length

Modnames	AICcWt	Estimate	SE
p: ~..time + bs(ls, df = 4) + ..time:bs(ls, df = 4)	0.85	60,485	12,635
p: ~..time + bs(ls, df = 5) + ..time:bs(ls, df = 5)	0.14	59,567	10,895
p: ~..time + ls + I(ls ^2) + ..time:ls + ..time:I(ls ^2)	0.00	59,210	9,366
p: ~-1 + I(as.numeric(..time == 1)) + I(as.numeric(..time == 2)):length.class	0.00	49,536	3,629
p: ~-1 + length.class:..time	0.00	60,614	13,039
p: ~-1 + I(as.numeric(..time == 2)) + I(as.numeric(..time == 1)):length.class	0.00	49,536	3,629
p: ~length.class + ..time	0.00	93,218	39,716
p: ~-1 + ..time	0.00	49,536	3,629
p: ~..time + ls + I(ls * 2)	0.00	49,563	3,635
Model averaged		60,351	12,398

The spline model is so much a better fit that it overwhelms the other models and the model averaged estimate is basically the first model. Notice that the model averaged confidence interval is larger than the original model because the variation in estimates among the models has also been taken into account.

The use of a spline fit is an alternative to the methods of Chen and LLoyd (2000) who used a non-parametric smoother to estimate the catchabilities at each sampling event. The advantage of the spline method is that the AICc framework can be used to rank the various models.

7.3 Non-parametric smoothing

Chen and Lloyd (2000) developed a non-parametric smoother for the relationship between a continuous covariate and the catchability at the sample events. The data are divided into bins, a stratified-Petersen estimator is used on each bin, and a smoother is used to avoid the capture probabilities or the abundance estimates from varying wildly between successive bins.

The default fit is obtained using:

```
# fit the Chen and Lloyd estimator using default bin width
data(data_NorthernPike)
nop.CL <- Petersen::LP_CL_fit(data_NorthernPike, covariate="length")
```

A plot of the recapture probabilities shows the upwards recapture probabilities with the larger lengths (Figure 16)

```
nop.CL$fit$plot1
```

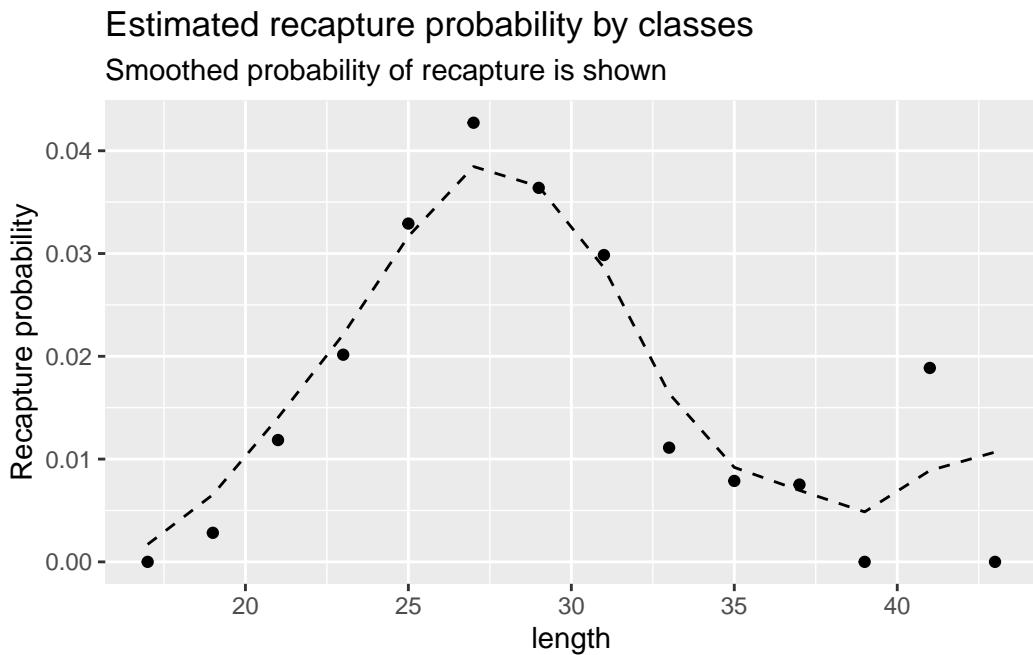


Figure 16: Estimated probabilities of recapture by length

And a plot of the abundance as a function of length along with the estimated overall abundance (Figure 17)

```

nop.CL$fit$plot2
nop.CL$summary
##   N_hat_f      N_hat_rn      N_hat N_hat_SE N_hat_conf_level N_hat_conf_method
## 1 ~1 (Intercept) 56620.94 5509.565                      0.95                  logN
##   N_hat_LCL N_hat_UCL p_model cond.ll n parms nobs      method
## 1 46789.66 68517.92       NA       NA       NA 7805 ChenLloyd

```

Estimated frequency distribution – estimated population size
SE for smoothed and raw estimates are shown in dashed lines at bottom

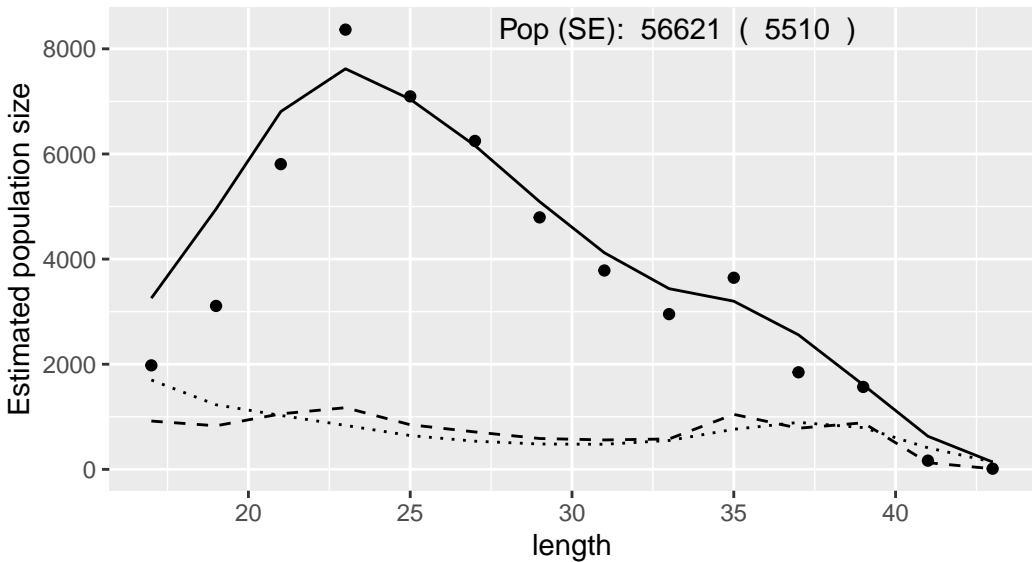


Figure 17: Estimated abundance of Northern Pike by length

The estimated abundance is comparable to the estimates from the using the semi-parametric spline method with a smaller standard error.

Because this is a non-parametric fit, it is not possible to compute a likelihood value and so AICc methods can not be used to compare this model with the previous models. However, the spline fit has a comparable flexibility and fits nicely into the AICc framework.

More bins leads to a “wigglier fit” and some playing with the smoothing parameters is needed to avoid spikes in abundance which are artefacts of the small sample sizes (especially recaptures) in the smaller bins seen around lengths of 40 inches or higher. Generally speaking, the smaller the bins, the larger the smoothing standard deviation and vice-verso.

```
# fit the Chen and Lloyd estimator with lower smoothing parameter
data(data_NorthernPike)
```

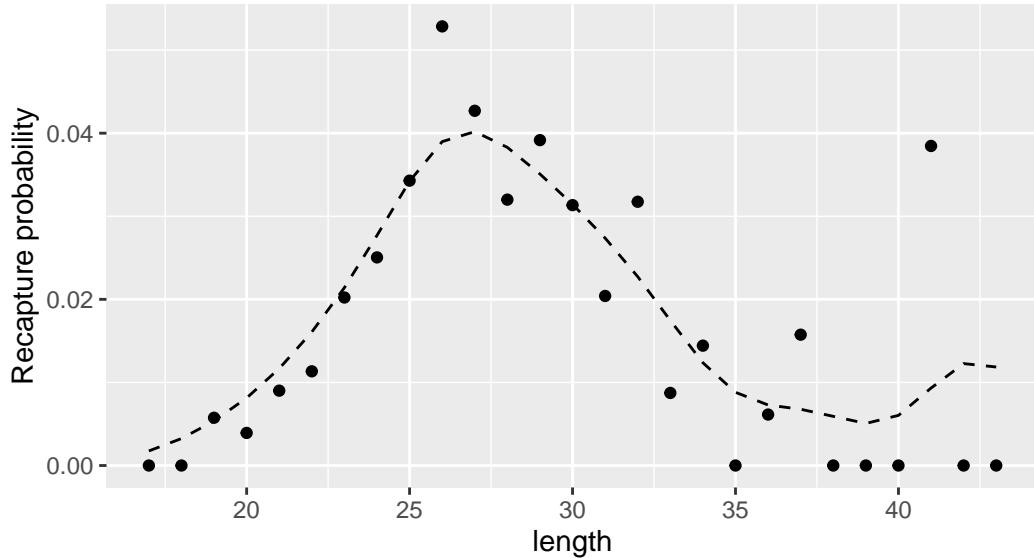
```

old.centers <- nop.CL$covar.data$center
nop.CL2 <- Petersen::LP_CL_fit(data_NorthernPike, covariate="length",
                                centers=seq(17, 43, 1), h1=1.5, h2=1.5)
nop.CL2$fit$plot1

```

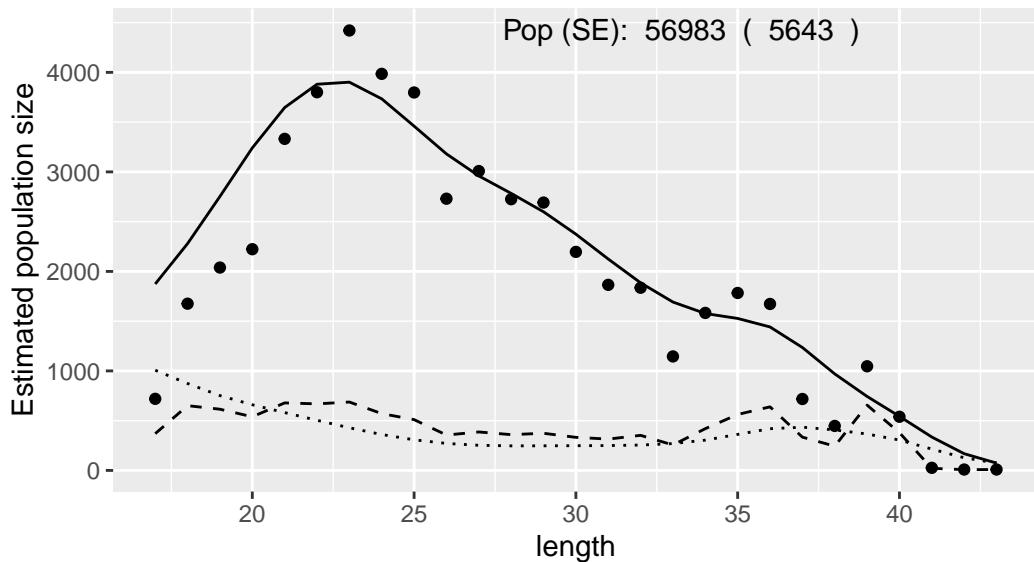
Estimated recapture probability by classes

Smoothed probability of recapture is shown



```
nop.CL2$fit$plot2
```

Estimated frequency distribution – estimated population size
 SE for smoothed and raw estimates are shown in dashed lines at bottom



```
nop.CL2$summary
##   N_hat_f      N_hat_rn      N_hat N_hat_SE N_hat_conf_level N_hat_conf_method
## 1 ~1 (Intercept) 56982.7 5642.788          0.95              logN
##   N_hat_LCL N_hat_UCL p_model cond.ll n parms nobs      method
## 1 46930.12 69188.58       NA        NA       NA 7805 ChenLloyd
```

8 Accounting for heterogeneity III - geographic or temporal stratification

8.1 Introduction

In some cases, heterogeneity in catchability at both sampling events is related to the location where the fish was sampled (i.e., geographical stratification) or the date when the fish was sampled (temporal stratification) or both.

Data consists of the number of fish releases in each of s release strata (n_{1s}), which are recaptured in one of t recapture strata, leading to an $s \times t$ recapture matrix (m_{ij}). Finally, there are n_{2t} unmarked fish captured for the first time in the second sampling event in the t recapture strata.

The key difference between these two cases is that in geographic stratification, it is theoretically possible to move from any one geographic stratum to any other geographic stratum leading

to a general movement matrix. However, in temporal stratification, you cannot capture the fish at the second sampling event before it is released, leading to recapture matrices that have zeros on the lower diagonal.

Schaefer (1951) developed an estimate of the total population abundance, N , using ratio and expectation arguments. The Schaefer estimate is biased unless capture probabilities are equal in all initial strata or the recovery probabilities are the same in all the final strata. If either condition holds, the pooled Petersen will also be unbiased and will be more precise (it makes more efficient use of the data). No estimate of standard error (s.e.) is available for the Schaefer estimate. **Consequently, the use of the Schaefer (1951) estimator is no longer recommended**

Seber (1982) summarizes the early work of Chapman and Junge (1956) and Darroch (1961). There are 3 cases:(a) $s = t$; (b) $s < t$; and (c) $s > t$. In case (a), the recapture matrix (m) is square, and a simple matrix-based estimator is available. Both the initial and final stratum (population) abundances can be estimated. This estimate is commonly referred to as the Darroch estimate. They gave the necessary and sufficient conditions for the pooled Petersen to be unbiased and developed two chi-square tests for sufficient conditions (i.e., if the tests fail to detect an effect, it may “safe” to pool...if tests detect an effect, it may or may not be safe to pool).These are the two χ^2 tests presented earlier.

In case (b), only the initial stratum (population) abundances can be estimated, and in case (c) only the final stratum abundances can be estimated. The total population size, N , is nevertheless estimable. Plante (1990) developed an alternate maximum likelihood method for the Darroch estimate that can be applied to all 3 cases including the cases where $s \neq t$.

A key issue with geographic stratification, is that it is very sensitive to singularities in the recapture matrix. For example, it fails if any row is a multiple of other rows, or a linear combination of other rows. It leads to estimates with very large standard errors (and nonsensical stratum estimates) if the recapture matrix is close to singularity.

This often requires pooling of rows or columns to reduce the singularity of the recapture matrix. Schwarz and Taylor (1998) review the estimators for geographic stratification and provides guidance on how to pool rows or columns. This data are also often sparse requiring much pooling. This is difficult to do in a structured fashion. If all rows are pooled to a single row, or all columns pooled to a single column, the estimator reduces to a Pooled-Petersen estimator.

Arnason et al (1996) created a Windows program (*SPAS*) to help in the analysis of geographical stratification. Schwarz (2023) ported the functionality to *R* and included “logical pooling” so that different poolings could be compared using AIC etc. The *Petersen* package provides a wrapper to make it easier to use. The *SPAS* software also includes an *autopool()* function.

While *SPAS* could also be used for temporal stratification, but the problem is that there is no way to use the “structural zeros” representing fish being captured before being released, or fish being recaptured more than a small number of weeks after being released. This gives a diagonal or band-diagonal structure to the temporally stratified data which *SPAS* ignores.

Bjorkstedt (2000). created the *DARR* (Darroch Analysis with Rank-Reduction) computer program to automatically pool the above sparse matrices but this program uses simple rules to ensure that adequate sample sizes are available in each of the release and recovery strata. It ignores the highly structured form of the data and cannot deal with many problems commonly encountered in such data such as missing strata. In response to this, Bonner and Schwarz (2011) developed *BTSPAS* which uses a Bayesian approach with a hierarchical model for the capture probabilities to share information when data are spares, and a spline for the run shape to again share information and to provide a straightforward way to interpolate over missing data. Again, a wrapper is provided in the *Petersen* package to simplify usage of *BTSPAS*.

8.2 Geographic stratification

8.2.1 Sampling protocol

Consider an study to estimate the number of return salmon to a river. The returns extends over several weeks and there are several spawning sites. As the adult salmon return, they are captured and marked with individually numbered tags and released at the first capture location using, for example, a fishwheel. The migration continues, and stream walks of the spawning grounds notes the number of tagged and untagged fish..

The efficiency of the fishwheels varies over time in response to stream flow, run size passing the wheel and other uncontrollable events. So it is unlikely that the capture probabilities are equal over time, i.e. are heterogeneous over time. Similarly, the effort at the spawning grounds varies by ground and is also heterogeneous.

This is a form of temporal x geographical stratification – the key feature in geographical stratification is that fish from any of the tagging strata, can, in theory, go to any of the recovery strata. There is no “natural” ordering of the geographical strata; the temporal strata have a natural ordering.

In the more general case, both strata are “geographically stratified”.

8.2.2 Data structure

The same data structure as previously seen is also used, except the capture history is modified to account for potentially many geographical or temporal strata as follows:

- $xx..yy$ represents a capture_history where xx and yy are the temporal or geographical stratum (e.g., julian week or spawning area) and ‘.’ separates the two strata from the two sampling event. If a fish is released in a stratum xx and never captured again, then yy is set to 0; if a fish is newly captured in stratum yy , then xx is set to zero.
- frequency variable for the number of fish with this capture history
- other covariates for the stratum yy .

It is not possible to model the initial capture probability or to separate fish by additional stratification variables in the *SPAS* software. These have rarely found to be useful in the geographically stratified models.

Some care is needed to properly account for 0 or missing values. Strata with 0 releases, should be dropped from the data structure. Similarly, strata with 0 recapture and 0 newly tagged fish should also be dropped. As shown later, some pooling of sparse rows/columns may be needed.

8.2.3 Key assumptions

The new key assumptions for this method are:

- all marked fish move to the recovery strata in the same proportions as unmarked fish in any release stratum
- there is no fall back of tagged fish, i.e., after marking the marked fish continue on their migration in same fashion as unmarked fish.
- catchability of marked fish represents the catchability of the unmarked fish

Unfortunately, for most of these assumptions, there is little information in the data that help detect violations. Gross violations of this model can be detected from the goodness-of-fit tests presented in Arnason et al. (1996) and the output from *SPAS* when the number of rows is not equal to the number of columns.

8.2.4 Harrison River Example

Returning salmon were captured and tags at a down river trap. A colored tag that varied by week of capture was applied. During the spawning season, stream walks took place at several spawning sites where the number of tags by color and the number of untagged fish was recorded on a weekly basis.

This is a combination of temporal stratification (when tags applied) and geographic stratification the spawning areas.

Here is part of the data:

```
data(data_spas_harrison)
head(data_spas_harrison, n=10)
```

	cap_hist	freq
1	01..00	130
2	02..00	330
3	03..00	790

```

4    04..00 667
5    05..00 309
6    06..00 65
7    00..a 744
8    01..a   4
9    02..a   12
10   03..a   7

```

Because we allow the stratum labels to be longer than a single digit, we use the “..” to separate strata labels in the capture history.

In week 1, 130 fish were tagged and never seen again; 4 fish were tagged in week 1 and recovered in area a , etc. This data can be summarized into a matrix:

		fw2					
fw1	00	a	b	c	d	e	f
00	0	744	1187	2136	951	608	127
01	130	4	2	1	1	0	0
02	330	12	7	14	1	3	0
03	790	7	11	41	9	1	1
04	667	1	13	40	12	9	1
05	309	0	1	8	8	3	0
06	65	0	0	0	0	0	1

The first row (corresponding to $fw1=0$) are the number of unmarked fish recovered in spawning area a . The first column (corresponding to $fw2=0$) are the number of released marked fish that were never seen again. The matrix in the lower bottom of the above array shows the number of fish released in each temporal stratum and recaptured in each spawning area.

8.2.5 Fitting the SPAS model

The SPAS model is fit to the above data using the wrapper supplied with this package. If a finer control is wanted on the fit, please refer to the documentation from the *SPAS* package. The very sparse last row will make fitting the model to the original data difficult with a very flat likelihood surface and so we relax the convergence criteria:

The `row.pool.in` and `col.pool.in` inform the function on which rows or columns to pool before the analysis proceeds. Both parameters use a vector of codes (length s for row pooling and length t for column pooling) where rows/columns are pooled that share the same value.

For example. `row.pool.in=c(1,2,3,4,5,6)` would imply that no rows are pooled, while `row.pool.in=c('a','a','a','b','b','b')` would imply that the first three rows and last

three rows are pooled. The entries in the vector can be numeric or character; however, using character entries implies that the final pooled matrix is displayed in the order of the character entries. I find that using entries such as '123' to represent pooling rows 1, 2, and 3 to be easiest to use.

The SPAS system only fits models were the number of rows after pooling is less than or equal to the number of columns after pooling.

8.2.6 Results from model 1 (no pooling).

The summary of the fit is:

```
harr..mod..1$summary
```

	p_model	name_model	cond.ll	n parms	nobs	method
1	Refer to row.pool/col.pool	No restrictions	47243.45	48	8256	SPAS
	cond.factor					
1	4121.288					

The usual conditional likelihood is presented etc. **Of more importance is the condition factor. This indicated how close to singularity the recovery matrix is with larger values indicating closer to singularity. Usually, this value should be 1000 or less to avoid numerical issues in the fit.**

The results of the model fit is a LARGE list. But the *SPAS.print.model()* function produces a nice report

```
SPAS::SPAS.print.model(harr..mod..1$fit)
```

```
Model Name: No restrictions
Date of Fit: 2025-10-24 19:00
Version of OPEN SPAS used : SPAS-R 2025.2.1
```

Raw data

	a	b	c	d	e	f	
01	4	2	1	1	0	0	130
02	12	7	14	1	3	0	330
03	7	11	41	9	1	1	790
04	1	13	40	12	9	1	667
05	0	1	8	8	3	0	309
06	0	0	0	0	0	1	65

744 1187 2136 951 608 127 0

Row pooling setup : 1 2 3 4 5 6
Col pooling setup : 1 2 3 4 5 6
Physical pooling : FALSE
Theta pooling : FALSE
CJS pooling : FALSE

Chapman estimator of population size 70135 (SE 4503)

Raw data AFTER PHYSICAL (but not logical) POOLING

	pool1	pool2	pool3	pool4	pool5	pool6	
pool.1	4	2	1	1	0	0	130
pool.2	12	7	14	1	3	0	330
pool.3	7	11	41	9	1	1	790
pool.4	1	13	40	12	9	1	667
pool.5	0	1	8	8	3	0	309
pool.6	0	0	0	0	0	1	65
	744	1187	2136	951	608	127	0

Condition number of XX' where X= (physically) pooled matrix is 4121.288

Condition number of XX' after logical pooling 4121.288

Large value of kappa (>1000) indicate that rows are approximately proportional which is not good

Conditional Log-Likelihood: 47243.45 ; np: 48 ; AICc: -94390.9

Code/Message from optimization is: 0 relative convergence (4)

Estimates

	pool1	pool2	pool3	pool4	pool5	pool6	psi	cap.prob	exp	factor
pool.1	3.7	2.6	0.8	0.9	0.0	0	130	0.005	191.2	
pool.2	12.0	7.0	14.0	1.0	3.0	0	330	1.000	0.0	
pool.3	7.0	11.0	41.0	9.0	1.0	1	790	1.000	0.0	
pool.4	1.0	13.8	37.5	11.8	10.9	1	667	0.021	47.6	
pool.5	0.0	1.0	7.7	7.9	3.3	0	309	0.037	26.3	
pool.6	0.0	0.0	0.0	0.0	0.0	1	65	0.012	79.4	
est unmarked	744.0	1186.0	2139.0	951.0	606.0	127	0	NA	NA	
	Pop	Est								
pool.1		26523								
pool.2		367								

pool.3	860
pool.4	36104
pool.5	8998
pool.6	5307
est unmarked	78159

SE of above estimates

	pool1	pool2	pool3	pool4	pool5	pool6	psi	cap.prob	exp	factor
pool.1	1.6	1.3	0.8	1.0	0.0	0	11.4	0.002		83.1
pool.2	3.5	2.6	3.7	1.0	1.7	0	18.2	0.000		0.0
pool.3	2.6	3.3	6.4	3.0	1.0	1	28.1	0.000		0.0
pool.4	1.0	3.6	5.9	3.5	2.1	1	25.8	0.007		16.7
pool.5	0.0	1.0	2.7	2.9	2.0	0	17.6	0.072		53.7
pool.6	0.0	0.0	0.0	0.0	0.0	1	8.1	0.015		94.7
est unmarked	NA	NA	NA	NA	NA	NA	0.0	NA		NA

Pop Est

pool.1	11462
pool.2	0
pool.3	0
pool.4	12411
pool.5	17661
pool.6	6253
est unmarked	14676

```

Chisquare gof cutoff : 0.1
Chisquare gof value : 0.843434
Chisquare gof df    : 0
Chisquare gof p     : NA

```

The original data, the data after pooling, estimates and their standard errors are shown. Here the stratified-Petersen estimate of the total number of fish passing the first sampling station is 78,159 with a standard error of 14,676.

The **N** entries refer to the population size; the **N.stratum** entries refer to the individual stratum population sizes; the **cap** entries refer to the estimated probability of capture in each row stratum; the **exp.factor** entries refer to $(1-\text{cap})/\text{cap}$, or the expansion factor for each row; the **psi** entries refer to the number of animals tagged but never seen again (the right most column in the input data); and the **theta** entries refer to the expected number of animals that were tagged in row stratum i and recovered in column stratum j (after pooling).

The fit is not entirely satisfactory because notice the very small population estimates for in weeks 2 and 3 or releases. This is unrealistic and is an indication of potential singularity problems in the data as noted with the large condition number seen earlier.

You can extract parts of the “printed” output by using the `extract=TRUE` argument in the `SPAS.print.model()` function:

```
# demonstrate how to extract part of output
harr..mod..1.res <- SPAS::SPAS.print.model(harr..mod..1$fit, extract=TRUE)
names(harr..mod..1.res)
```

```
[1] "model_name" "date"          "version"      "input"        "Chapman"
[6] "fit"         "spas"          "gof"
```

```
harr..mod..1.res$spas$estimate
```

	pool1	pool2	pool3	pool4	pool5	pool6	psi	cap.prob	exp	factor
pool.1	3.7	2.6	0.8	0.9	0.0	0	130	0.005	191.2	
pool.2	12.0	7.0	14.0	1.0	3.0	0	330	1.000	0.0	
pool.3	7.0	11.0	41.0	9.0	1.0	1	790	1.000	0.0	
pool.4	1.0	13.8	37.5	11.8	10.9	1	667	0.021	47.6	
pool.5	0.0	1.0	7.7	7.9	3.3	0	309	0.037	26.3	
pool.6	0.0	0.0	0.0	0.0	0.0	1	65	0.012	79.4	
est unmarked	744.0	1186.0	2139.0	951.0	606.0	127	0	NA	NA	
	Pop	Est								
pool.1	26523									
pool.2	367									
pool.3	860									
pool.4	36104									
pool.5	8998									
pool.6	5307									
est unmarked	78159									

```
harr..mod..1.res$spas$se
```

	pool1	pool2	pool3	pool4	pool5	pool6	psi	cap.prob	exp	factor
pool.1	1.6	1.3	0.8	1.0	0.0	0	11.4	0.002	83.1	
pool.2	3.5	2.6	3.7	1.0	1.7	0	18.2	0.000	0.0	
pool.3	2.6	3.3	6.4	3.0	1.0	1	28.1	0.000	0.0	
pool.4	1.0	3.6	5.9	3.5	2.1	1	25.8	0.007	16.7	
pool.5	0.0	1.0	2.7	2.9	2.0	0	17.6	0.072	53.7	
pool.6	0.0	0.0	0.0	0.0	0.0	1	8.1	0.015	94.7	
est unmarked	NA	NA	NA	NA	NA	NA	0.0	NA	NA	
	Pop	Est								

```

pool.1      11462
pool.2        0
pool.3        0
pool.4     12411
pool.5     17661
pool.6      6253
est unmarked 14676

```

```

cat("\n\nEstimated total abundance ", harr..mod..1.res$spas$estimate[nrow(harr..mod..1.res$spas$estimate),
                                                               ncol(harr..mod..1.res$spas$estimate)
                                                               "(SE ", harr..mod..1.res$spas$se[nrow(harr..mod..1.res$spas$estimate),
                                                               ncol(harr..mod..1.res$spas$estimate)], ") fish", "\n")

```

```
Estimated total abundance 78159 (SE 14676 ) fish
```

```
rm(harr..mod..1.res) # not need further
```

8.2.7 Pooling some rows and columns

As noted by Darroch (1961), the stratified-Petersen will fail if the matrix of movements is close to singular. This often happens if two rows are proportional to each other. In this case, there is no unique MLE for the probability of capture in the two rows, and they should be pooled. A detailed discussion of pooling is found in Schwarz and Taylor (1998).

There is no simple way to determine which rows/column to pool but when estimated population sizes are small for a stratum, this is usual an indication that some pooling is required. The *SPAS* system has an **experimental** autopool feature which may serve as a guide.

8.2.7.1 Pooling pairs of rows

Let us now pool the first two rows, the next two rows, and the last two rows,

The code is

```

harr..mod..2 <- Petersen::LP_SPAS_fit(data_spas_harrison, model.id="Pooling every second row"
                                         row.pool.in=c("12","12","34","44","56","56"),
                                         col.pool.in=c(1,2,3,4,5,6))

```

```
Using nlmminb to find conditional MLE
outer mgc: 1967.318
outer mgc: 1585.784
outer mgc: 601.797
outer mgc: 361.6262
outer mgc: 255.3465
outer mgc: 301.4938
outer mgc: 25.14831
outer mgc: 2.82591
outer mgc: 2.424709
outer mgc: 16.30222
outer mgc: 4.565925
outer mgc: 24.77543
outer mgc: 57.44612
outer mgc: 7.92634
outer mgc: 11.63666
outer mgc: 10.76265
outer mgc: 3.863489
outer mgc: 13.76323
outer mgc: 26.44275
outer mgc: 4.706843
outer mgc: 23.51753
outer mgc: 0.8778394
outer mgc: 5.614518
outer mgc: 1.738815
outer mgc: 5.426214
outer mgc: 16.44758
outer mgc: 1.748598
outer mgc: 1.453996
outer mgc: 6.546109
outer mgc: 6.563646
outer mgc: 8.584827
outer mgc: 20.45095
outer mgc: 2.996709
outer mgc: 1.732739
outer mgc: 5.660383
outer mgc: 6.150282
outer mgc: 6.135996
outer mgc: 10.84269
outer mgc: 2.233833
outer mgc: 5.003054
outer mgc: 2.24289
outer mgc: 2.801219
```

```

outer mgc:  0.6797359
outer mgc:  0.7992463
outer mgc:  0.06792093
outer mgc:  0.1106809
outer mgc:  0.04946835
outer mgc:  0.02579226
outer mgc:  0.01303348
outer mgc:  0.006503766
outer mgc:  0.003219488
outer mgc:  0.001356174
outer mgc:  0.001004277
outer mgc:  0.0005343734
outer mgc:  0.00013112
Convergence codes from nlminb  1 singular convergence (7)
Finding conditional estimate of N

```

```
#harr..mod..2$summary
```

Notice how we specify the pooling for rows and columns and the choice of entries for the two corresponding vectors. Now the condition factor is much better (well less than 1000).

The results of the model fit is

```

Model Name: Pooling every second row
Date of Fit: 2025-10-24 19:00
Version of OPEN SPAS used : SPAS-R 2025.2.1

```

Raw data

	a	b	c	d	e	f	
01	4	2	1	1	0	0	130
02	12	7	14	1	3	0	330
03	7	11	41	9	1	1	790
04	1	13	40	12	9	1	667
05	0	1	8	8	3	0	309
06	0	0	0	0	0	1	65
	744	1187	2136	951	608	127	0

```

Row pooling setup : 12 12 34 44 56 56
Col pooling setup : 1 2 3 4 5 6
Physical pooling : FALSE
Theta pooling     : FALSE
CJS pooling       : FALSE

```

Chapman estimator of population size 70135 (SE 4503)

Raw data AFTER PHYSICAL (but not logical) POOLING

	pool1	pool2	pool3	pool4	pool5	pool6	
pool.12	4	2	1	1	0	0	130
pool.12	12	7	14	1	3	0	330
pool.34	7	11	41	9	1	1	790
pool.44	1	13	40	12	9	1	667
pool.56	0	1	8	8	3	0	309
pool.56	0	0	0	0	0	1	65
	744	1187	2136	951	608	127	0

Condition number of XX' where X= (physically) pooled matrix is 4121.288

Condition number of XX' after logical pooling 189.4612

Large value of kappa (>1000) indicate that rows are approximately proportional which is not good

Conditional Log-Likelihood: 47242.37 ; np: 46 ; AICc: -94392.74

Code/Message from optimization is: 1 singular convergence (7)

Estimates

	pool1	pool2	pool3	pool4	pool5	pool6	psi	cap.prob	exp	factor
pool.12	3.5	2.8	0.9	1	0.0	0.0	130	0.019		52.1
pool.12	10.4	10.0	12.4	1	3.1	0.0	330	0.019		52.1
pool.34	7.0	11.0	41.0	9	1.0	1.0	790	1.000		0.0
pool.44	0.9	15.3	37.5	12	9.1	1.1	667	0.036		26.6
pool.56	0.0	1.6	6.9	8	3.1	0.0	309	0.015		66.2
pool.56	0.0	0.0	0.0	0	0.0	1.5	65	0.015		66.2
est unmarked	746.0	1180.0	2141.0	951	608.0	126.0	0	NA		NA

Pop Est

pool.12	7327
pool.12	19484
pool.34	860
pool.44	20470
pool.56	22123
pool.56	4438
est unmarked	74701

SE of above estimates

	pool1	pool2	pool3	pool4	pool5	pool6	psi	cap.prob	exp	factor
pool.12	1.8	1.9	0.9	1.0	0.0	0.0	11.4	0.006		15.6
pool.12	3.3	3.0	3.2	1.0	1.5	0.0	18.2	0.006		15.6
pool.34	2.6	3.3	6.4	3.0	1.0	1.0	28.1	0.000		0.0
pool.44	0.9	4.4	6.6	3.4	2.7	1.1	25.8	0.021		16.0
pool.56	0.0	1.7	2.3	2.6	1.4	0.0	17.6	0.009		39.5
pool.56	0.0	0.0	0.0	0.0	0.0	0.7	8.1	0.009		39.5
est unmarked	NA	NA	NA	NA	NA	NA	0.0	NA		NA
	Pop	Est								
pool.12		2149								
pool.12		5715								
pool.34		0								
pool.44		11887								
pool.56		13007								
pool.56		2609								
est unmarked		10284								

```

Chisquare gof cutoff : 0.1
Chisquare gof value : 2.857233
Chisquare gof df    : 2
Chisquare gof p     : 0.2396402

```

Here the stratified-Petersen estimate of the total number of smolts passing the first sampling station is 74,701 with a standard error of 74,701. which is a slight reduction from the unpooled estimates.

The estimates look better, but a standard error of 0 for the abundance in some strata is an indication that the fit is not very realistic.

8.2.7.2 Pooling to early vs late.

Let us now pool the first three rows and the last three rows,

The code is

```

harr..mod..3 <- Petersen::LP_SPAS_fit(data_spas_harrison, model.id="Pooling early vs. late",
                                         row.pool.in=c("123","123","123","456","456","456"),
                                         col.pool.in=c(1,2,3,4,5,6))

```

```

Using nlminb to find conditional MLE
outer mgc: 2986.733

```

```
outer mgc: 1769.526
outer mgc: 816.8674
outer mgc: 269.1311
outer mgc: 30.10549
outer mgc: 24.07398
outer mgc: 5.393645
outer mgc: 18.96964
outer mgc: 2.953744
outer mgc: 1.395084
outer mgc: 18.87373
outer mgc: 6.688677
outer mgc: 1.640811
outer mgc: 7.332616
outer mgc: 1.685392
outer mgc: 7.317009
outer mgc: 1.522422
outer mgc: 6.294555
outer mgc: 1.261084
outer mgc: 4.980596
outer mgc: 2.103006
outer mgc: 8.561154
outer mgc: 1.79518
outer mgc: 6.70074
outer mgc: 2.203129
outer mgc: 8.175411
outer mgc: 2.210091
outer mgc: 7.714625
outer mgc: 3.180497
outer mgc: 11.01373
outer mgc: 4.797201
outer mgc: 18.68681
outer mgc: 10.12453
outer mgc: 12.11322
outer mgc: 6.303981
outer mgc: 6.197803
outer mgc: 2.248121
outer mgc: 0.7088488
outer mgc: 0.5721706
outer mgc: 0.1888875
outer mgc: 0.07641965
outer mgc: 0.02903072
outer mgc: 0.01084322
outer mgc: 0.004013247
```

```

outer mgc: 0.001470131
outer mgc: 0.0001335366
outer mgc: 9.781857e-06
Convergence codes from nlminb 1 singular convergence (7)
Finding conditional estimate of N

```

```
#harr..mod..3$summary
```

Notice how we specify the pooling for rows and columns and the choice of entries for the two corresponding vectors. Now the condition factor is again much better (well less than 1000).

The results of the model fit is

```

Model Name: Pooling early vs. late
Date of Fit: 2025-10-24 19:00
Version of OPEN SPAS used : SPAS-R 2025.2.1

```

Raw data

	a	b	c	d	e	f	
01	4	2	1	1	0	0	130
02	12	7	14	1	3	0	330
03	7	11	41	9	1	1	790
04	1	13	40	12	9	1	667
05	0	1	8	8	3	0	309
06	0	0	0	0	0	1	65
	744	1187	2136	951	608	127	0

```

Row pooling setup : 123 123 123 456 456 456
Col pooling setup : 1 2 3 4 5 6
Physical pooling : FALSE
Theta pooling : FALSE
CJS pooling : FALSE

```

```
Chapman estimator of population size 70135 (SE 4503 )
```

```

Raw data AFTER PHYSICAL (but not logical) POOLING
pool1 pool2 pool3 pool4 pool5 pool6
pool.123     4      2      1      1      0      0 130
pool.123    12      7     14      1      3      0 330
pool.123     7     11     41      9      1      1 790

```

pool.456	1	13	40	12	9	1	667
pool.456	0	1	8	8	3	0	309
pool.456	0	0	0	0	0	1	65
	744	1187	2136	951	608	127	0

Condition number of \mathbf{XX}' where \mathbf{X} = (physically) pooled matrix is 4121.288
 Condition number of \mathbf{XX}' after logical pooling 22.90096

Large value of kappa (>1000) indicate that rows are approximately proportional which is not good

Conditional Log-Likelihood: 47237.51 ; np: 44 ; AICc: -94387.01

Code/Message from optimization is: 1 singular convergence (7)

Estimates

	pool1	pool2	pool3	pool4	pool5	pool6	psi	cap.prob	exp	factor
pool.123	4.8	2.5	0.8	1.1	0.0	0.0	130	0.038		25.4
pool.123	14.4	8.8	10.9	1.1	3.9	0.0	330	0.038		25.4
pool.123	8.4	13.9	31.9	9.8	1.3	1.4	790	0.038		25.4
pool.456	1.2	17.1	30.1	13.2	12.1	1.5	667	0.033		29.2
pool.456	0.0	1.3	6.0	8.8	4.0	0.0	309	0.033		29.2
pool.456	0.0	0.0	0.0	0.0	0.0	1.5	65	0.033		29.2
est unmarked	739.0	1177.0	2160.0	948.0	603.0	126.0	0	NA		NA
	Pop	Est								
pool.123		3642								
pool.123		9685								
pool.123		22695								
pool.456		22445								
pool.456		9939								
pool.456		1994								
est unmarked		70399								

SE of above estimates

	pool1	pool2	pool3	pool4	pool5	pool6	psi	cap.prob	exp	factor
pool.123	2.3	1.7	0.8	1.1	0.0	0.0	11.4	0.007		4.8
pool.123	3.8	3.1	2.9	1.1	2.2	0.0	18.2	0.007		4.8
pool.123	3.0	3.7	4.7	3.0	1.3	1.3	28.1	0.007		4.8
pool.456	1.3	3.9	4.4	3.2	2.9	1.2	25.8	0.006		5.8
pool.456	0.0	1.3	2.1	2.8	2.1	0.0	17.6	0.006		5.8
pool.456	0.0	0.0	0.0	0.0	0.0	1.2	8.1	0.006		5.8
est unmarked	NA	NA	NA	NA	NA	NA	0.0	NA		NA
	Pop	Est								
pool.123		664								

```

pool.123      1767
pool.123      4140
pool.456      4299
pool.456      1904
pool.456      382
est unmarked   4552

```

```

Chisquare gof cutoff : 0.1
Chisquare gof value  : 12.99086
Chisquare gof df     : 4
Chisquare gof p      : 0.01132054

```

Here the stratified-Petersen estimate of the total number of smolts passing the first sampling station is 70,399 with a standard error of 70,399. which is a slight reduction from the unpooled estimates.

The estimates seem more sensible.

8.2.7.3 Pooling to a single row and complete pooling

You can pool to a single row (and multiple columns) or a single row and single column both of which are equivalent to the pooled Petersen estimator. The code and output follow:

```

Using nlmnb to find conditional MLE
outer mgc: 5554.604
outer mgc: 3255.445
outer mgc: 686.8353
outer mgc: 85.66413
outer mgc: 12.0596
outer mgc: 3.803003
outer mgc: 0.6639956
outer mgc: 0.2609515
outer mgc: 0.09867538
outer mgc: 0.03679076
outer mgc: 0.01360926
outer mgc: 0.005017272
outer mgc: 0.001859472
outer mgc: 0.0002086563
outer mgc: 1.313641e-05
Convergence codes from nlmnb 1 singular convergence (7)
Finding conditional estimate of N

```

Model Name: A single row
Date of Fit: 2025-10-24 19:00
Version of OPEN SPAS used : SPAS-R 2025.2.1

Raw data

	a	b	c	d	e	f	
01	4	2	1	1	0	0	130
02	12	7	14	1	3	0	330
03	7	11	41	9	1	1	790
04	1	13	40	12	9	1	667
05	0	1	8	8	3	0	309
06	0	0	0	0	0	1	65
	744	1187	2136	951	608	127	0

Row pooling setup : 1 1 1 1 1 1
Col pooling setup : 1 2 3 4 5 6
Physical pooling : FALSE
Theta pooling : FALSE
CJS pooling : FALSE

Chapman estimator of population size 70135 (SE 4503)

Raw data AFTER PHYSICAL (but not logical) POOLING

	pool1	pool2	pool3	pool4	pool5	pool6	
pool.1	4	2	1	1	0	0	130
pool.1	12	7	14	1	3	0	330
pool.1	7	11	41	9	1	1	790
pool.1	1	13	40	12	9	1	667
pool.1	0	1	8	8	3	0	309
pool.1	0	0	0	0	0	1	65
	744	1187	2136	951	608	127	0

Condition number of XX' where X= (physically) pooled matrix is 4121.288
Condition number of XX' after logical pooling 1

Large value of kappa (>1000) indicate that rows are approximately proportional which is not good

Conditional Log-Likelihood: 47237.43 ; np: 43 ; AICc: -94388.87

Code/Message from optimization is: 1 singular convergence (7)

Estimates

	pool1	pool2	pool3	pool4	pool5	pool6	psi	cap.prob	exp	factor
pool.1	4.5	2.6	0.8	1.1	0.0	0.0	130	0.036		27.1
pool.1	13.6	8.9	10.7	1.1	4.2	0.0	330	0.036		27.1
pool.1	8.0	14.0	31.4	10.1	1.4	1.5	790	0.036		27.1
pool.1	1.1	16.6	30.6	13.5	12.5	1.5	667	0.036		27.1
pool.1	0.0	1.3	6.1	9.0	4.2	0.0	309	0.036		27.1
pool.1	0.0	0.0	0.0	0.0	0.0	1.5	65	0.036		27.1
est unmarked	741.0	1178.0	2160.0	947.0	602.0	125.0	0	NA		NA
	Pop	Est								
pool.1		3883								
pool.1		10326								
pool.1		24198								
pool.1		20906								
pool.1		9257								
pool.1		1857								
est unmarked		70426								

SE of above estimates

	pool1	pool2	pool3	pool4	pool5	pool6	psi	cap.prob	exp	factor
pool.1	2.1	1.8	0.8	1.1	0.0	0.0	11.4	0.002		1.9
pool.1	3.0	3.1	2.8	1.1	2.2	0.0	18.2	0.002		1.9
pool.1	2.6	3.6	4.4	2.9	1.3	1.3	28.1	0.002		1.9
pool.1	1.1	3.8	4.4	3.2	2.9	1.3	25.8	0.002		1.9
pool.1	0.0	1.3	2.1	2.8	2.2	0.0	17.6	0.002		1.9
pool.1	0.0	0.0	0.0	0.0	0.0	1.3	8.1	0.002		1.9
est unmarked	NA	NA	NA	NA	NA	NA	0.0	NA		NA
	Pop	Est								
pool.1		262								
pool.1		696								
pool.1		1632								
pool.1		1410								
pool.1		624								
pool.1		125								
est unmarked		4545								

```

Chisquare gof cutoff : 0.1
Chisquare gof value  : 13.09369
Chisquare gof df     : 5
Chisquare gof p      : 0.0225164

```

Now the estimated abundance of 70,426 with a standard error of 4,545. which is a slight

reduction from the unpooled estimates.

8.2.7.4 Comparing the different poolings:

We can compare the fit using AICc in the usual way (Table 23)

```
spas.aictab <- Petersen::LP_AICc(harr..mod..1, harr..mod..2, harr..mod..3, harr..mod..4)
```

Table 23: Comparision of model different SPAS poolings

Model specification	Conditional log-likelihood	# parms	# obs	AICc	Delta	AICcWt
Pooling every second row	47,242.37	46	8,256	-94,392.22	0.00	0.63
No restrictions	47,243.45	48	8,256	-94,390.32	1.89	0.24
A single row	47,237.43	43	8,256	-94,388.41	3.81	0.09
Pooling early vs. late	47,237.51	44	8,256	-94,386.53	5.69	0.04

Notice that even with a high degree of logical pooling, **SPAS** still estimates a movement probability for the entire recapture matrix. All that logical pooling does is enforce equality of the initial capture probabilities. It is possible to do physical row pooling where the recapture matrix is also reduced, but then you cannot compare different physical poolings using AIC.

We see that there is a slight preference for pooling every two rows vs no pooling, but the condition factor of the model with no pooling makes it less than ideal.

We can also create a report of the estimates and model average them in the usual way (Table 24).

Table 24: Model averaged estimates of abundance from different SPAS poolings

Modnames	AICcWt	Estimate	SE
Pooling every second row	0.63	74,701	10,284
No restrictions	0.24	78,159	14,676
A single row	0.09	70,426	4,545
Pooling early vs. late	0.04	70,399	4,552

Modnames	AICcWt	Estimate	SE
Model averaged	74,986	11,252	

Different column pooling will give the same fit and so cannot be compared. Refer to the vignette in the *SPAS* package on why column poolings cannot be compared for more details.

8.2.7.5 What does autopool suggest?

We can also try the autopooling options where *SPAS* pools rows and columns to ensure that sufficient data is present, but, (at this moment) does not check the singularity condition.

The autopooling function only suggests pooling the last 2 rows because the sample sizes are very small, but the condition factor is still large and so I would not consider this model.

8.2.8 Not all tagged fish have tags read in recovery strata

Schwarz, Andrews, and Link (1999) considered with model with the additional complexity that some of the fish recovered are seen to have tags, but the tags are not read. Consequently, it is not possible to distribute these fish back to the release stratum. In their example, fish pass through a fishway where it relatively easy to see if a fish has a tag, but only every 1/4 of fish are subsequently removed to read the tag and know the stratum of release.

Specialized software to fit the above model using the estimating equations has been written in *R* and is available upon request. This software also will fit simpler models where the tag-reading rate is assumed to be equal for all the recovery strata, where the tag application rate is equal for all release strata, or where the parameters can be modeled using covariates. In the middle model above, the estimate of the total population size is algebraically equal to the simple Petersen estimator, which is known to be consistent when all the initial capture probabilities are equal.

In the absence of specialized software for this experiment, approximate solutions can be found using software for the stratified Petersen with all tags read (*SPAS*) by initially distributing the unread tags into their respective rows in the same proportion as the m_{ij} to their column sums. Of course, the final variance estimates reported by *SPAS* will be too small.

8.2.9 Summary

The greatest problem that you will likely encounter while using *SPAS* is the near singularity of the recovery matrix. This is often unavoidable because there is no simple way to simplify the structure of the model and so all $s \times t$ possible movement patterns must be accounted for. If the stratification is temporal, this imposes much additional structure that can be used

(see next section). Consequently, I suspect you will find that more than 3 or 4 rows or columns is about the limit that can successfully fit with *SPAS* in practice.

8.3 Temporal stratification

Temporal stratification is commonly used when estimating the run abundance of incoming adult salmon or outgoing juvenile salmon. The key difference from geographical stratification is that it is not possible for fish to be captured before they are released leading to a large number of structural 0's in the counts. The incoming/out coming abundance generally changes 'slowly' over time, so that the abundance in temporal stratum i has information on temporal stratum $i + 1$. Problems with the data are common, e.g. no fish released in some of the temporal strata, or no operations in other temporal strata, e.g., due to safety concerns leading to structural zeros.

Bonner and Schwarz (2011) developed a Bayesian model for temporally stratified sampling and created an *R* package, *BTSPAS* that analyses these studies. In this package, we provide a wrapper function to make it easier to use the *BTSPAS* package and to present results in a unified fashion. However, if finer control is wanted over the fitting procedures, the *BTSPAS* package can be used directly. Details about this model are presented in Bonner and Schwarz (2011) and in the vignettes that ship with the package. A synopsis of the theory is presented in Appendix D.

8.3.1 Sampling Protocol

Consider a study to estimate the number of outgoing smolts on a small river. The run of smolts extends over several weeks. As smolts migrate, they are captured and marked with individually numbered tags and released at the first capture location using, for example, a fishwheel. The migration continues, and a second fishwheel takes a second sample several kilometers down stream. At the second fishwheel, the captures consist of a mixture of marked (from the first fishwheel) and unmarked fish.

The efficiency of the fishwheels varies over time in response to stream flow, run size passing the wheel and other uncontrollable events. So it is unlikely that the capture probabilities are equal over time at either location, i.e. are heterogeneous over time.

We suppose that we can temporally stratify the data into, for example, weeks, where the capture-probabilities are (mostly) homogeneous at each wheel in each week.

8.3.2 Data structure

The same data structure as previously seen is also used, except the capture history is modified to account for potentially many temporal strata as follows:

- $xx..yy$ represents a capture_history where xx and yy are the temporal stratum (e.g., julian week) and ‘.’ separates the two temporal strata. If a fish is released in temporal stratum and never captured again, then yy is set to 0; if a fish is newly captured in temporal stratum yy , then xx is set to zero.
- frequency variable for the number of fish with this capture history
- other covariates for this temporal stratum yy .

It is not possible to model the initial capture probability or to separate fish by stratification variables. These have rarely found to be useful in the temporally stratified models.

Some care is needed to properly account for 0 or missing values. For temporal strata with 0 releases, the BTSPAS wrappers will automatically impute 0 for the number of releases and recaptures if no capture history records are present. However, no imputation is done for strata is missing u_2 values because this could represent a case were the trap was not running, rather than the trap did not capture any unmarked fish. The summary statistics should be carefully checked when using this wrappers.

8.3.3 Key assumptions

The new key assumptions for this method are:

- sampling at second station is for the entire period of the stratum It is untrue, the user may need may need to impute and deal with the additional uncertainty
- all marked fish move at same rate as unmarked fish
- there is no fall back of tagged fish, i.e., after marking the marked fish continue on their migration in same fashion as unmarked fish.
- catchability of marked fish represents the catchability of the unmarked fish

Unfortunately, for most of these assumptions, there is little information in the data that help detect violations. Gross violations of this model can be detected from the goodness-of-fit plots presented in Bonner and Schwarz (2011).

There are two common cases

- Fish are stratified into temporal strata, and fish released in temporal stratum i , move together and are captured together in a single future temporal stratum. This is called the **Diagonal Model**.
- Fish are stratified into temporal strata, but fish released in temporal stratum i can be captured in a number of future temporal strata. This is the called the **NonDiagonal Case**

The *BTSPAS* package also has other functions for dealing with multiple age classes in the study, but these are not discussed in this document.

8.3.4 Diagonal Model

In the diagonal model, fish in a marked-cohort tend to move together and so tend to be recaptured close in time as well. Suppose that fish captured and marked in each week tend to migrate together so that they are captured in a single subsequent stratum. For example, suppose that in each Julian week j , n_{1j} fish are marked and released above the rotary screw trap. Of these, m_{2j} are recaptured. All recaptures take place in the week of release, i.e. the matrix of releases and recoveries is diagonal. The n_{1j} and m_{2j} establish the capture efficiency of the second trap in Julian week j .

At the same time, u_{2j} unmarked fish are captured at the screw trap.

We label the releases as n_{1i} to indicate this is the “first” capture and release of a cohort of fish, and m_{2i} or u_{2i} to indicate that this is “second” event where tagged fish are recaptured, and unmarked fish are newly captured.

Here is an example of data collected under this protocol:

```
data(data_btspas_diag1)
head(data_btspas_diag1)
```

	cap_hist	freq	logflow
1	00..04	14587	6.564691
2	04..00	1414	6.564691
3	04..04	51	6.564691
4	00..05	2854	7.077220
5	05..00	1189	7.077220
6	05..05	146	7.077220

In temporal stratum 4, 1465 fish were released at the first fishwheel, of which 51 were recaptured in the same week at the second fishwheel, and 1414 were never seen again. An additional 14587 unmarked fish were captured at the second fishwheel.

This can be summarized in a matrix format (up to week 10):

		fw2	0	4	5	6	7	8	9	10	11	12
fw1			0	14587	2854	1027	1945	2855	1323	933	57549	14846
0	0		1414	51	0	0	0	0	0	0	0	0
4	4		1189	0	146	0	0	0	0	0	0	0
5	5		180	0	0	17	0	0	0	0	0	0
6	6		1167	0	0	0	168	0	0	0	0	0
7	7		1581	0	0	0	0	72	0	0	0	0

9	880	0	0	0	0	43	0	0	0
10	582	0	0	0	0	0	28	0	0
11	7672	0	0	0	0	0	0	297	0
12	5695	0	0	0	0	0	0	0	313

The first row (corresponding to $fw1=0$) are the number of unmarked fish recovered at the second fish wheel in the temporal strata. The first column (corresponding to $fw2=0$) are the number of released marked fish that were never seen again.

We can also get the summary statistics (n_1 , m_2 , and u_2) for this data:

```
# compute n, m,u
nmu <- Petersen::cap_hist_to_n_m_u(data_btspas_diag1)
temp <- data.frame(time=nmu$..ts, n1=nmu$n1, m2=nmu$m2, u2=nmu$u2)
temp
```

	time	n1	m2	u2
1	4	1465	51	14587
2	5	1335	146	2854
3	6	197	17	1027
4	7	1335	168	1945
5	8	1653	72	2855
6	9	923	43	1323
7	10	610	28	933
8	11	7969	297	57549
9	12	6008	313	14846
10	13	3770	151	5291
11	14	4854	309	9249
12	15	3350	376	4615
13	16	1062	110	1433
14	17	346	40	446
15	18	88	12	120
16	19	20	1	23

There are no missing values

The theory in Appendix D, assumes that a single spline curve can be fit across all strata, but problems can arise in fitting this model to the data. In some cases, there are obvious breaks in the pattern of abundance over time that need to be accounted for in the model. F Attempting to fit a smooth curve across all strata ignores these jumps and so it is necessary to allow for breaks in the fitted spline.

It may occur that no marked fish (i.e., $n_{1i} = 0$ for some i) are released in a particular stratum because no fish were available or because of logistical constraints. This would imply that a simple Petersen estimator for this stratum cannot be computed because no estimate of the capture probability is available. However, the Bayesian method will impute a range of capture-probabilities based on the capture probabilities in other strata, the shape of the spline curve, and the observed number of unmarked fish captured. The final estimate of abundance will (automatically) incorporate the uncertainty for this imputed capture probability.

If no data could be collected for a particular stratum (i.e., all of $n_{1i} = 0$, $m_{2i} = 0$, and $u_{2i} = 0$), the spline will “impute” a value for the run size and capture probability in that stratum given the shape of the spline and the variability in individual run sizes about the spline; and will “impute” a value for the capture probability given the range of capture probabilities in the other strata. The final estimate of abundance will (automatically) incorporate the uncertainty for this imputed values.

While it is possible to interpolate for several strata in a row, there is of course, no information on the shape of the underlying spline during these missed strata, and so the results should be interpreted with care.

The Bayesian model assumes that sampling occurs throughout the temporal stratum. However, in some strata, sampling may take place in part of the stratum (i.e., in 3 of 7 days in a week). This causes no theoretical problem for estimation of the capture probability as the capture probability is assumed to be equal over the entire week so estimates of the probability of capture at the second trap based on 3 days may have poor precision but remain unbiased. However, the number of unmarked fish needs to be adjusted for the partial sampling during the stratum. We assume that the user has made this adjustment, and no accounting of the uncertainty in this adjustment will be made.

In some cases, the recapture effort varies over the course of a temporal stratum. For example, two rotary screw traps may be operating on Monday, and then only one trap is operating on Tuesday, etc. Unfortunately, this type of problem CANNOT be adequately dealt with by the spline (or any other method) that uses batch marks. The problem is that differing effort during a stratum (week) results in heterogeneity of catchability during the week, e.g., the catchability on days when two traps are operating is likely larger than the catchability on days when only one trap is operating. If a batch mark is used, the data is pooled over the stratum (week) and it is impossible to separate out catches according to how many traps are operating.

As for the pooled-Petersen estimator, this will likely result in estimates with low bias, but the precision of the estimates for these strata will be mis-reported, i.e., the standard deviations of the estimated run sizes for these strata will be understated. While there is no way to assess the extent of the problem (other than via simulations), it is hoped that the stratification into weekly strata will resolve most of the underreporting of the precision by the pooled-Petersen estimator and that any remaining understatement is not material.

The two key advantages of the Bayesian approach are:

- accounting for missing data when sampling could not take place
- the self-adjusting performance of the method. If sample sizes are large in each temporal stratum, then very little sharing of information is needed across the strata and very little smoothing of the run distribution is done. However, in case with small sample sizes, more sharing of information across strata occurs and more smoothing is done on the run shape.

Full details are presented in Schwarz and Bonner (2011) and the vignettes of the *BTSPAS* package.

8.3.4.1 Preliminary screening of the data

A pooled-Petersen estimator would add all of the marked, recaptured and unmarked fish to give an estimate of 1,987,456 (SE 41,322) fish but can the estimate be trusted?

Let us first examine a plot of the estimated capture efficiency at the second trap for each set of releases (Figure 18).

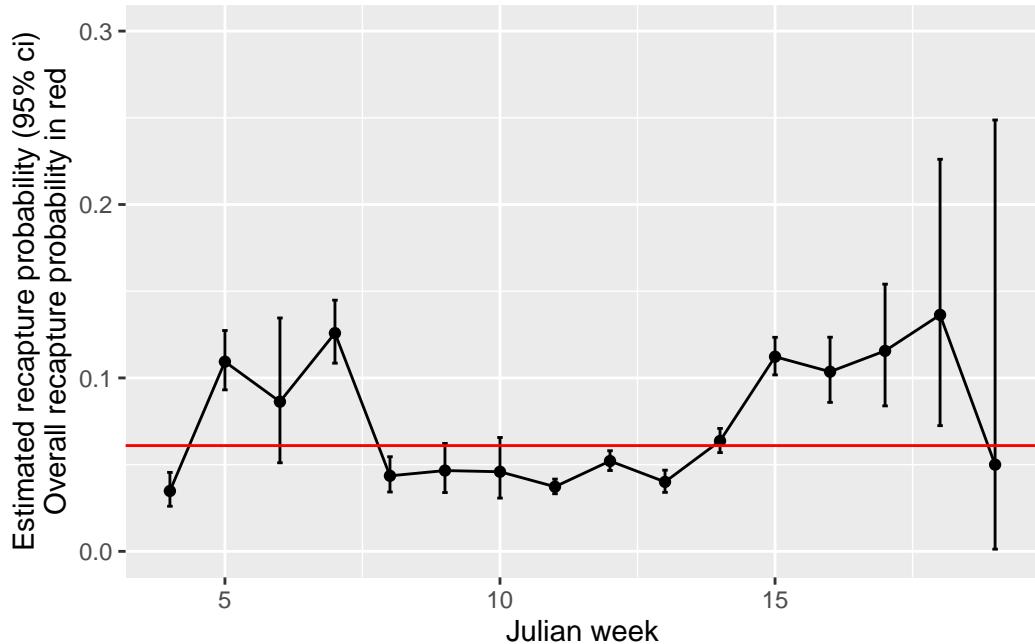


Figure 18: Empirical capture probabilities

There are several unusual features

- There appears to be heterogeneity in the capture probabilities across the season.
- In some temporal strata, the number of marked fish released and recaptured is very small which lead to estimates with poor precision.

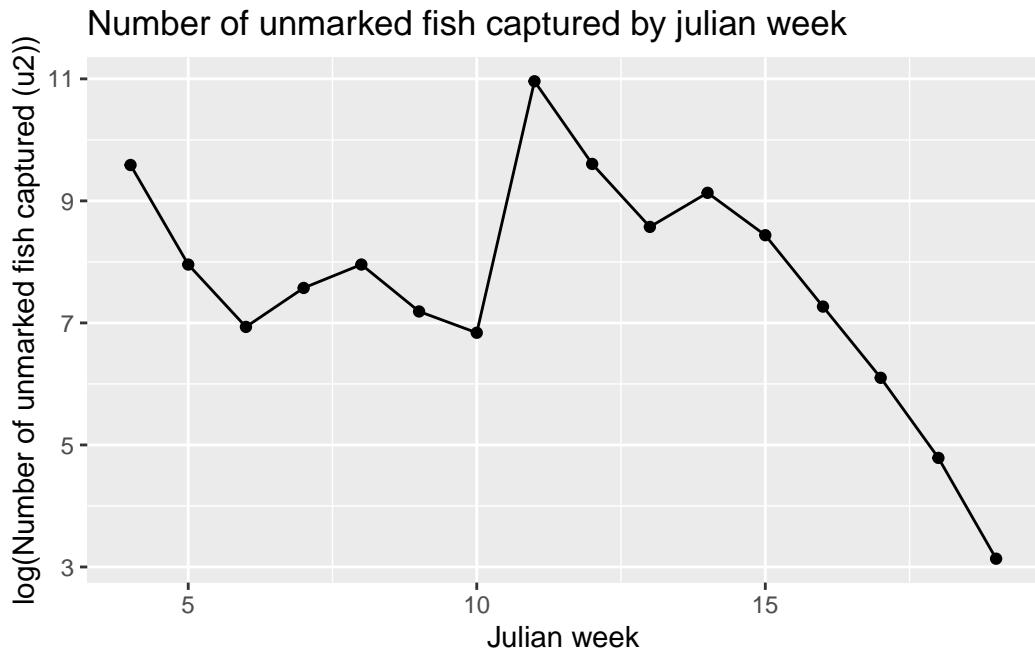


Figure 19: Observed number of unmarked recaptures

Similarly, let us look at the pattern of unmarked fish captured at the second trap (Figure 19).

The number of unmarked fish captured suddenly jumps by several orders of magnitude (remember the above plot is on the $\log()$ scale) in temporal stratum 11. This jump corresponds to releases of hatchery fish into the system.

Finally, let us look at the individual estimates for each stratum found by computing a Petersen estimator for abundance of unmarked fish for each individual stratum (Figure 20).

We see:

- The sudden jumps in abundance due to the hatchery releases is apparent
- There is a fairly regular pattern in abundance with a slow increase until the first hatchery release, followed by a steady decline.

8.3.4.2 Fitting the basic BTSPAS diagonal model

The *BTSPAS* package attempts to strike a balance between the completely pooled Petersen estimator and the completely stratified Petersen estimator. In the former, capture probabilities are assumed to equal for all fish in all strata, while in the latter, capture probabilities are allowed to vary among strata in no structured way. Furthermore, fish populations often have a general structure to the run, rather than arbitrarily jumping around from stratum to stratum.

The *BTSPAS* package also has additional features and options:

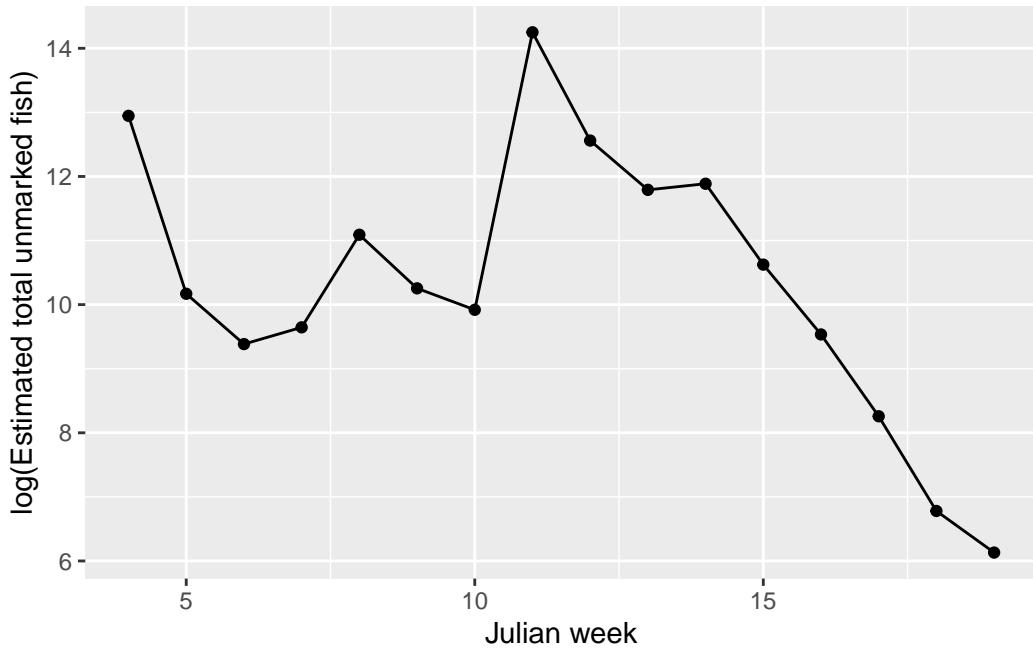


Figure 20: Estimated $\log(\text{total unmarked})$ by julian week

- the user to use covariates to explain some of the variation in the capture probabilities.
- if u_2 is missing for any stratum, the program will use the spline to interpolate the number of unmarked fish in the population for the missing stratum.
- if n_1 **and** m_2 are 0, then these strata provide no information towards recapture probabilities. This is useful when no release take place in a stratum (e.g. trap did not run) and so you need ‘dummy’ values as placeholders. Of course if $n_1 > 0$ and $m_2 = 0$, this provides information that the capture probability may be small. If $n_1 = 0$ and $m_2 > 0$, this is an error (recoveries from no releases).
- the program allows you specify break points in the underlying spline to account for external events. We was in the above example, that hatchery fish were released at in julian weeks 23 and 40 resulting in sudden jump in abundance. The *jump.after* parameter gives the julian weeks just BEFORE the sudden jump, i.e. the spline is allowed to jump AFTER the julian weeks in *jump.after*.

If unfortunate events happen where, for example, no fish could be released, or the second fish wheel is not running in a week, the data should be carefully modified to ensure that missing values are not replaced by 0’s.

The *Petersen* package function *BTSPAS_Diag_fit()* is a wrapper to the corresponding function in the *BTSPAS* package that takes the (modified for bad events) data file, a model for the mean capture probabilities at the second temporal stratum (default is a common mean), and identification of break points in the underlying spline, and then call the function in the *BTSPAS*

package, and formats the returned data structure to match the returning structure for other functions in this package.

If finer control over the fitting process is needed, the *BTSPAS* package should be called directly – please consult the vignettes that come with the *BTSPAS* package.

We fit the model and look at the summary information on the fit:

```
BTSPAS.diag.fit1 <- Petersen::LP_BTSPAS_fit_Diag(  
  data_btspas_diag1,  
  p_model=~1,  
  jump.after=10,  
  InitialSeed=23943242  
)
```

The output object contains all of the results and can be saved for later interrogations. This is useful if the run takes considerable time (e.g. overnight) and you want to save the results for later processing.

As noted previously, model comparisons are not easily done using Bayesian methods except for perhaps using DIC (but this is often computed at the incorrect focus of the model). Furthermore, the BTSPAS model are self adjusting and usually the default fit is sufficient.

The final BTSPAS object returned by the fit has many components and contain summary tables, plots, and the like:

```
names(BTSPAS.diag.fit1)
```

```
[1] "summary"      "data"          "p_model"       "p_model_cov"  "jump.after"  
[6] "InitialSeed" "name_model"   "fit"           "datetime"
```

```
names(BTSPAS.diag.fit1$fit)
```

```
[1] "n.chains"        "n.iter"         "n.burnin"  
[4] "n.thin"          "n.keep"         "n.sims"  
[7] "sims.array"      "sims.list"      "sims.matrix"  
[10] "summary"         "mean"          "sd"  
[13] "median"          "root.short"    "long.short"  
[16] "dimension.short" "indexes.short" "last.values"  
[19] "program"         "model.file"    "isDIC"  
[22] "DICbyR"          "pV"            "DIC"  
[25] "time2run"        "pD"            "DIC2"  
[28] "model"           "parameters.to.save" "plots"  
[31] "runTime"         "report"        "data"
```

The *plots* sub-object contains many plots:

```
[1] "init.plot"          "fit.plot"          "logitP.plot"
[4] "acf.Utot.plot"     "post.UNtot.plot"   "gof.plot"
[7] "trace.logitP.plot" "trace.logU.plot"
```

In particular, it contains plots of the initial spline fit (*init.plot*), the final fitted spline (*fit.plot*), the estimated capture probabilities (on the logit scale) (*logitP.plot*), plots of the distribution of the posterior sample for the total unmarked and marked fish (*post.UNtot.plot*) and model diagnostic plots (goodness of fit (*gof.plot*), trace (*trace...plot*), and autocorrelation plots (*act.Utot.plot*)).

These plots are all created using the *ggplot2* packages, so the user can modify the plot (e.g. change titles etc).

The *BTSPAS* program also creates a report, which includes information about the data used in the fitting, the pooled- and stratified-Petersen estimates, a test for pooling, and summaries of the posterior. Only the first few lines are shown below:

```
[1] "Time Stratified Petersen with Diagonal recaptures and error in smoothed U - Fri Oct 2
[2] "Version: 2024-11-01 "
[3] ""
[4] ""
[5] ""
[6] "  Results "
[7] ""
[8] "*** Raw data *** "
[9] "    time   n1   m2   u2 logitPcov[1]"
[10] " [1,]    4 1465  51 14587           1"
[11] " [2,]    5 1335 146  2854           1"
[12] " [3,]    6  197  17  1027           1"
[13] " [4,]    7 1335 168 1945           1"
[14] " [5,]    8 1653  72  2855           1"
[15] " [6,]    9  923  43 1323           1"
[16] " [7,]   10  610  28   933           1"
[17] " [8,]   11 7969 297 57549          1"
[18] " [9,]   12 6008 313 14846          1"
[19] " [10,]  13 3770 151  5291          1"
[20] " [11,]  14 4854 309  9249          1"
```

The fitted spline curve to the number of unmarked fish available in each recovery sample is shown in Figure 21

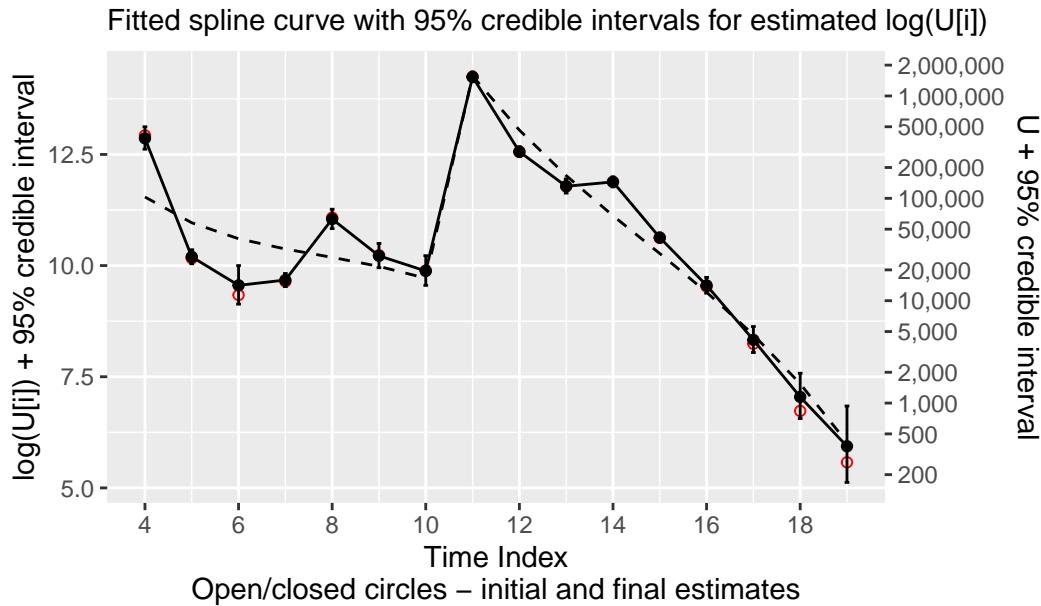


Figure 21: Fitted spline curve in BTSPAS diagonal model

The jump in the spline when hatchery fish are released is evident. The actual number of unmarked fish is allowed to vary around the spline as shown below.

The distribution of the posterior sample for the total number unmarked and total abundance is available as shown in Figure 22

A plot of the $\logit(P)$ (the logit of the estimated probability of capture) is shown in Figure 23

A summary of the posterior for each parameter is also available. The estimates of total abundance can be extracted and summarized in a similar fashion as in the other models:

```
BTSPAS.diag.est1 <- Petersen::LP_BTSPAS_est (BTSPAS.diag.fit1)
BTSPAS.diag.est1$summary
```

```
N_hat_f N_hat_rn   N_hat N_hat_SE N_hat_conf_level N_hat_conf_method
1       NA        NA 2752479 103391.3           0.95      Posterior
N_hat_LCL N_hat_UCL p_model name_model cond.ll n parms    nobs
1  2557827  2960623     ~1      p: ~1      NA      NA 154081
```

Or, all of the parameters can be extracted directly from the BTSPAS object as shown below for the total abundance and the total unmarked abundance.

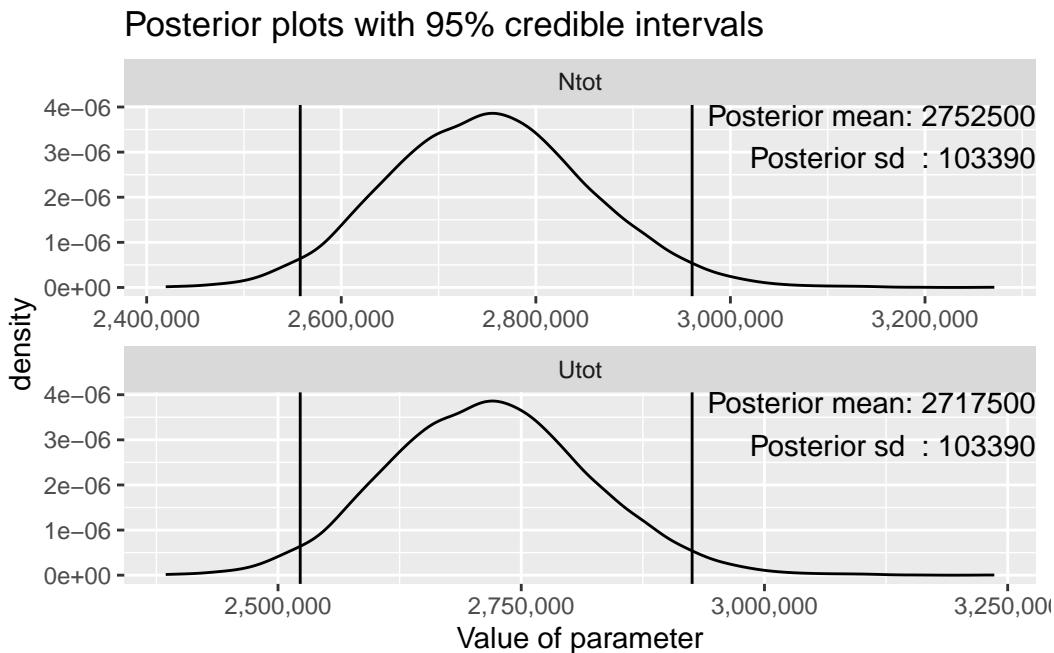


Figure 22: Distribution of posterior samples for total number unmarked and total abundance from BTSPAS diagonal model

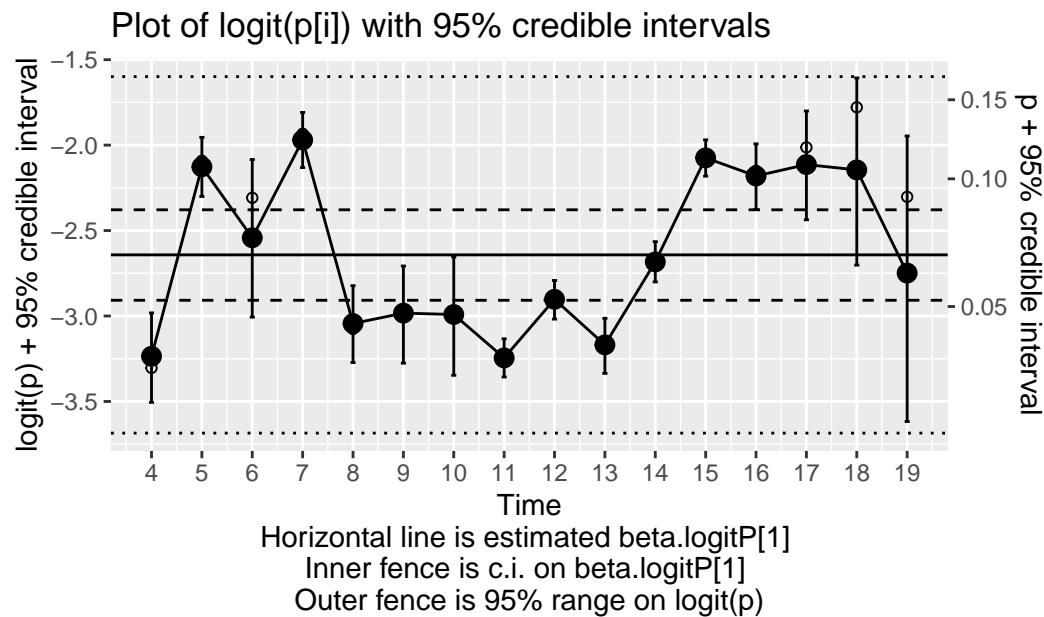


Figure 23: Plot of estimated $\text{logit}(P)$ from BTSPAS diagonal model

```
BTSPAS.diag.fit1$fit$summary[
  row.names(BTSPAS.diag.fit1$fit$summary) %in% c("Ntot", "Utot"),]
```

	mean	sd	2.5%	25%	50%	75%	97.5%	Rhat	n.eff
Ntot	2752479	103391.3	2557827	2681428	2750959	2819728	2960623	1.001657	2200
Utot	2717494	103391.3	2522842	2646443	2715974	2784743	2925638	1.001656	2200

This also includes the Rubin-Brooks-Gelman statistic (*Rhat*) on mixing of the chains and the effective sample size of the posterior (after accounting for autocorrelation).

The estimated total abundance from *BTSPAS* is 2,752,479 (SD 103,391) fish.

Samples from the posterior are also included in the *sims.matrix*, *sims.array* and *sims.list* elements of the BTSPAS results object.

It is always important to do model assessment before accepting the results from the model fit. Please check the vignettes of BTSPAS on details on how to interpret the goodness of fit, trace, and autocorrelation plots.

8.3.4.3 Dealing with problems in the data

The BTSPAS package is quite flexible when dealing with problems in the data. Please consult the vignettes with the BTSPAS package for many examples.

8.3.4.3.1 Missing data from some strata

In some cases, data is missing for some temporal strata. For example, river flows could be too high to use the fishwheels safely; crews are sick; data are lost; etc. In these cases, the spline is used to interpolate the run abundance over the missing data.

For example, we will delete data from temporal strata 5 and 8. The resulting summary data is:

```
xtabs( freq ~ fw1 + fw2,
       data=data_btspas_diag3.aug[ data_btspas_diag3.aug$fw1 %in% temporal_strata[1:10] &
                                     data_btspas_diag3.aug$fw2 %in% temporal_strata[1:10] , ,
       exclude=NULL, na.action=na.pass)
```

fw1	fw2	0	4	6	7	9	10	11	12	13	14
0	0	14587	1027	1945	1323	933	57549	14846	5291	9249	
4	1414	51	0	0	0	0	0	0	0	0	0

6	180	0	17	0	0	0	0	0	0	0
7	1167	0	0	168	0	0	0	0	0	0
9	880	0	0	0	43	0	0	0	0	0
10	582	0	0	0	0	28	0	0	0	0
11	7672	0	0	0	0	0	297	0	0	0
12	5695	0	0	0	0	0	0	313	0	0
13	3619	0	0	0	0	0	0	0	151	0
14	4545	0	0	0	0	0	0	0	0	309

Notice that there is no data for temporal strata 5 or 8.

We can also get the summary statistics (n_1 , m_2 , and u_2) for this data (and notice the warning messages):

```
# compute n, m,u
nmu <- Petersen::cap_hist_to_n_m_u(data_btspas_diag3)
```

Warning in Petersen::cap_hist_to_n_m_u(data_btspas_diag3): *** Caution... Missing value for n

Warning in Petersen::cap_hist_to_n_m_u(data_btspas_diag3): *** Caution... Missing value for u

```
temp <- data.frame(time=nmu$..ts, n1=nmu$n1, m2=nmu$m2, u2=nmu$u2)
temp
```

	time	n1	m2	u2
1	4	1465	51	14587
2	5	0	0	NA
3	6	197	17	1027
4	7	1335	168	1945
5	8	0	0	NA
6	9	923	43	1323
7	10	610	28	933
8	11	7969	297	57549
9	12	6008	313	14846
10	13	3770	151	5291
11	14	4854	309	9249
12	15	3350	376	4615
13	16	1062	110	1433
14	17	346	40	446
15	18	88	12	120
16	19	20	1	23

Notice that in stratum 5 and 8, the number of releases (n_1) and recaptures (m_2) are set to 0 (no data present), but the u_2 values are set to *NA* (missing) indicating that the number of unmarked fish recaptured is unknown because the trap was not running.

We fit the BTSPAS model in the same way as previously.

```
BTSPAS.diag.fit3 <- Petersen::LP_BTSPAS_fit_Diag(
  data_btspas_diag3,
  p_model=~1,
  jump.after=10,
  InitialSeed=234234
)
```

which gives the model fit summary:

The fitted spline curve to the number of unmarked fish available in each recovery sample is shown in Figure 24

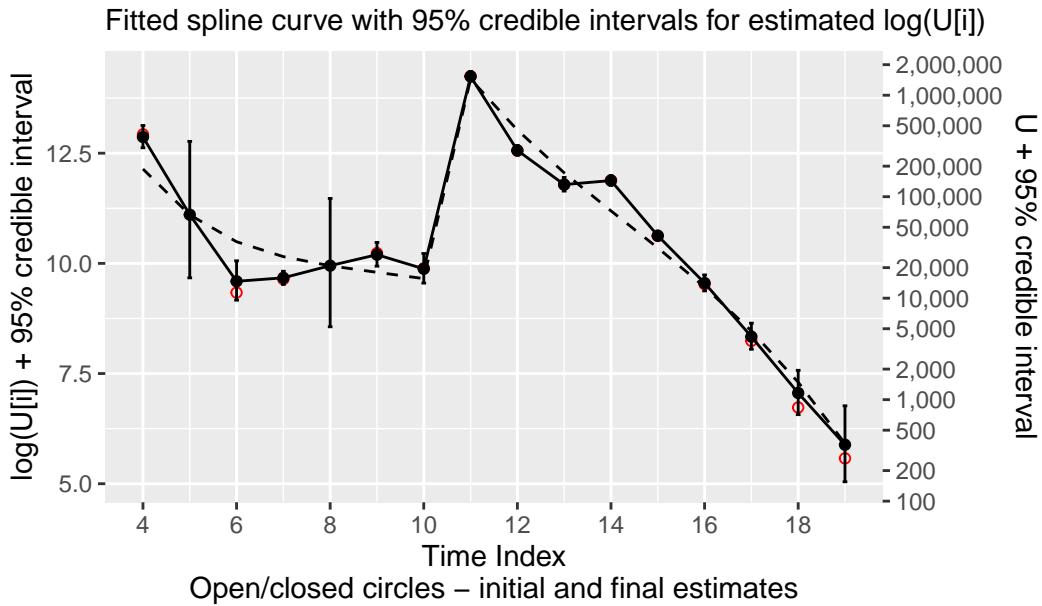


Figure 24: Fitted spline curve in BTSPAS diagonal model with missing data

Notice that the curve was interpolated at temporal strata 5 and 8 (with a much larger uncertainty) compared to the other temporal strata.

A plot of the $\logit(P)$ (the logit of the estimated probability of capture) is shown in Figure 25

In those cases with no data, the capture-probability is estimated from the hierarchical model (at the mean of the model) with high uncertainty, but this not used to estimate U_2 because u_2 is missing.

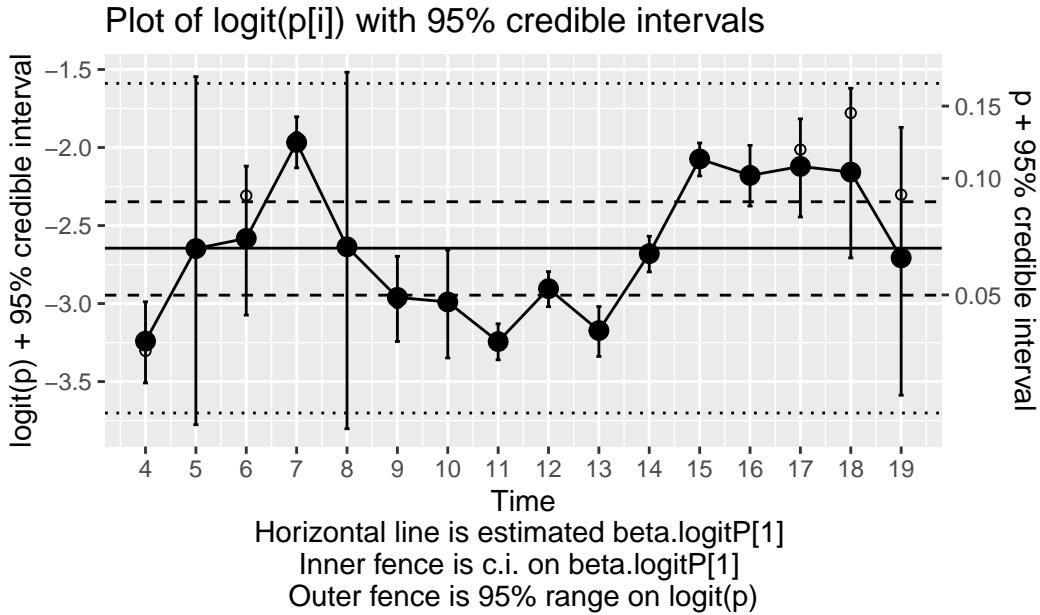


Figure 25: Plot of estimated $\logit(P)$ from BTSPAS diagonal model with missing data

A summary of the posterior for each parameter is also available. The estimates of total abundance can be extracted and summarized in a similar fashion as in the other models. The estimate is similar to the case where there was no missing data, but the uncertainty is larger.

```
BTSPAS.diag.est3 <- Petersen::LP_BTSPAS_est (BTSPAS.diag.fit3)
BTSPAS.diag.est3$summary
```

	N_{hat}_f	N_{hat}_{rn}	N_{hat}	N_{hat}_{SE}	$N_{\text{hat}}_{\text{conf_level}}$	$N_{\text{hat}}_{\text{conf_method}}$	
1	NA	NA	2780360	163253.3	0.95	Posterior	
	N_{hat}_{LCL}	N_{hat}_{UCL}	p_{model}	$\text{name}_{\text{model}}$	cond.11	n_{parms}	n_{obs}
1	2547384	3116045	-1	$p: \sim 1$	NA	NA	145384

8.3.4.3.2 Fixing p's

In some cases, the second trap is not running and so there are no recaptures of tagged fish and no captures of untagged fish. This usually ends up with 0's for the total number of untagged fish captured in a temporal stratum, even though fish were released.

We need to set the p's in these strata to 0 rather than letting *BTSPAS* impute a value based on the hierarchical model for the p's.

This is done by passing the temporal stratum where the $\logit(p)$ should be fixed typically at $\logit(p) = -10$ which corresponds to $p = 0.0000454$.

For example, to fix the $\text{logit}(p)$ for temporal stratum 12, the following code would be used:

```
BTSPAS.diag.fit2 <- Petersen::LP_BTSPAS_fit_Diag(
  data_btspas_diag1,
  p_model=~1,
  jump.after=10,
  logitP.fixed=12, logitP.fixed.values=-10,
  InitialSeed=23943242
)
```

8.3.4.3.3 Using covariates to model the p's

BTSPAS also allows you to model the p's with additional covariates, such as temperature, stream flow, etc. It is not possible to use covariates to model the total number of unmarked fish. A separate data frame must be created with the covariate value for each of the temporal strata at the second wheel.

Here is the data frame with the covariate data. There must be a line for every temporal stratum between the first and last stratum, even those where the traps were not running.

```
..ts2 logflow
2      4 6.564691
3      5 7.077220
4      6 6.884975
5      7 6.984033
6      8 7.923348
7      9 8.072650
8     10 7.817110
9     11 7.681441
10    12 7.607920
11    13 7.438067
12    14 6.391612
13    15 6.276237
14    16 6.464555
15    17 7.159880
16    18 6.559354
17    19 6.113682
```

A preliminary plot of the empirical logit shows an approximate quadratic fit to $\text{log}(flow)$, but the uncertainty in each week is enormous! (Figure 26)

The summary statistics are:

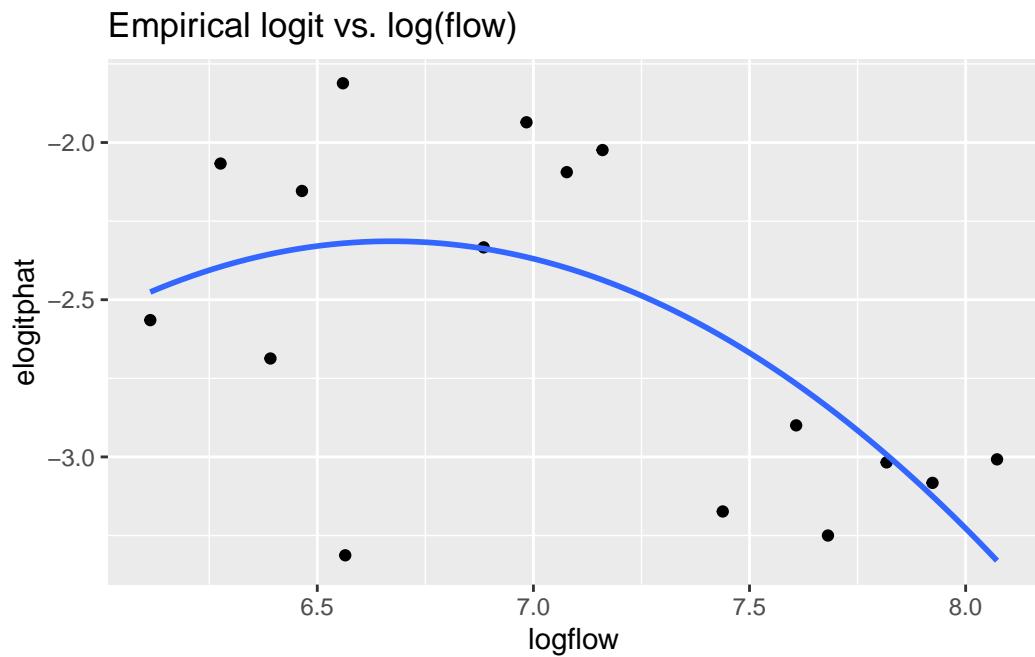


Figure 26: Logit(p) vs log(flow)

We can also get the summary statistics (n_1 , m_2 , and u_2) for this data:

	time	n1	m2	u2
1	4	1465	51	14587
2	5	1335	146	2854
3	6	197	17	1027
4	7	1335	168	1945
5	8	1653	72	2855
6	9	923	43	1323
7	10	610	28	933
8	11	7969	297	57549
9	12	6008	313	14846
10	13	3770	151	5291
11	14	4854	309	9249
12	15	3350	376	4615
13	16	1062	110	1433
14	17	346	40	446
15	18	88	12	120
16	19	20	1	23

There is no missing data.

We fit the model by modifying the formula for *p_model* and including the covariate data

```
BTSPAS.diag.fit4 <- Petersen::LP_BTSPAS_fit_Diag(  
  data_btspas_diag1,  
  p_model=~logflow+I(logflow^2), p_model_cov=p_cov,  
  jump.after=10,  
  InitialSeed=23943242  
)
```

Then the estimates can be extracted in the usual fashion:

```
BTSPAS.diag.est4 <- Petersen::LP_BTSPAS_est (BTSPAS.diag.fit4)  
BTSPAS.diag.est4$summary
```

```
N_hat_f N_hat_rn  N_hat N_hat_SE N_hat_conf_level N_hat_conf_method  
1      NA        NA 2741477 102175.2          0.95           Posterior  
N_hat_LCL N_hat_UCL      p_model      name_model  
1 2549871 2948895 ~logflow + I(logflow^2) p: ~logflow + I(logflow^2)  
cond.ll n parms   nobs  
1      NA        NA 154081
```

There is only a minor change in the estimate compared to not using a covariate (seen previously)

8.3.5 Non-diagonal case

In this case, the released fish are not recaptured in a single future temporal stratum, and recaptures takes place for a number of future strata. Here is an example of data collected under this protocol. Fish were tagged and released on the Conne River each day (denoted by the julian day of the year). The fish move upstream and are recaptured at an upstream trap (along with untagged fish).

```
data(data_btspas_nondiag1)  
data_btspas_nondiag1[ grepl("^027", data_btspas_nondiag1$cap_hist) |  
  grepl("000..027", data_btspas_nondiag1$cap_hist),]
```

```
cap_hist freq  
5 027..000 224  
20 000..027 1420  
28 027..028 1
```

34	027..029	39
40	027..030	84
47	027..031	35
54	027..032	1
61	027..033	3
68	027..034	1

In temporal stratum 27, 388 fish were released at the first fishwheel, of which 1 were recaptured in the next temporal stratum at the second fishwheel, 39 were recaptured in the following temporal stratum at the second fishwheel etc, and 224 were never seen again. An additional 1420 unmarked fish were captured at the second fishwheel.

Recaptures of marked fish no longer take place along the “diagonal” that was seen in the previous example (only the first 10 strata are shown below):

fw2		0	23	24	25	26	27	28	29	30	31	32	33
fw1		0	0	0	0	0	1420	27886	9284	4357	11871	14896	9910
0	0	0	0	0	0	0	1420	27886	9284	4357	11871	14896	9910
23	21	0	0	0	0	0	0	11	2	0	0	0	0
24	132	0	0	0	0	0	3	118	18	6	3	0	0
25	328	0	0	0	0	0	1	149	76	33	16	3	1
26	294	0	0	0	0	0	0	42	163	69	34	1	0
27	224	0	0	0	0	0	0	1	39	84	35	1	3
28	100	0	0	0	0	0	0	0	1	29	60	14	5
29	266	0	0	0	0	0	0	0	0	3	82	84	19
30	291	0	0	0	0	0	0	0	0	0	20	154	56
31	191	0	0	0	0	0	0	0	0	0	0	16	44

The first row (corresponding to $fw1=0$) are the number of unmarked fish recovered at the second fish wheel in the temporal strata. The first column (corresponding to $fw2=0$) are the number of released marked fish that were never seen again.

Most fish seem to take between 4 to 10 weeks to move between fish wheels.

Note that the second fishwheel was not operating in temporal strata 23 to 26, so no recoveries are possible, and the number of unmarked fish recaptured is also 0. The capture probability will have to be set to 0 for these temporal strata.

The summary statistics are:

```
# compute n, m,u
nmu <- Petersen::cap_hist_to_n_m_u(data_btspas_nondiag1)
temp <- data.frame(time =nmu$..ts,
```

```

n1  =c(nmu$n1,   rep(NA, length(nmu$u2)-length(nmu$n1))),  

m2  =rbind(nmu$m2, matrix(NA, nrow=length(nmu$u2)-length(nmu$n1), ncol=nmu$u2))  

u2  =nmu$u2)  

temp

```

	time	n1	m2.1	m2.2	m2.3	m2.4	m2.5	m2.6	m2.7	m2.8	m2.9	m2.10	u2
1	23	34	0	0	0	0	0	11	2	0	0	0	0
2	24	280	0	0	0	3	118	18	6	3	0	0	0
3	25	607	0	0	1	149	76	33	16	3	1	0	0
4	26	603	0	0	42	163	69	34	1	0	0	0	0
5	27	388	0	1	39	84	35	1	3	1	0	0	1420
6	28	212	0	1	29	60	14	5	2	0	1	0	27886
7	29	468	0	3	82	84	19	11	2	1	0	0	9284
8	30	586	0	20	154	56	53	9	1	1	1	0	4357
9	31	512	0	16	44	146	83	25	3	3	0	1	11871
10	32	458	0	2	52	125	82	13	14	3	0	0	14896
11	33	479	0	0	83	133	43	12	5	2	0	0	9910
12	34	329	0	12	96	60	29	8	0	1	0	0	16526
13	35	248	0	19	72	37	17	7	0	0	0	0	17443
14	36	201	0	7	55	23	6	1	0	0	0	0	16485
15	37	104	0	7	3	3	0	0	0	0	0	0	6776
16	38	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	4644
17	39	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	2190
18	40	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	1066
19	41	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	166

8.3.5.1 Preliminary screening of the data

A pooled-Petersen estimator would add all of the marked, recaptured and unmarked fish to give an estimate of 283,200 (SE 3,616) fish but can the estimate be trusted?

Let us first examine a plot of the estimated capture efficiency at the second trap for each set of releases (Figure 27) formed by looking at the total number of recoveries / total fish released.

There are several unusual features

- There appears to be heterogeneity in the total capture probabilities across the season with a lower catchability early in the season and a higher catchability later in the season
- The fall off in catchability near the end of the season is a artefact of sampling ending while fish are still migrating.

Similarly, let us look at the pattern of unmarked fish captured at the second trap (Figure 28).

Again notice no untagged recoveries in weeks 23 to 26 and the large jump in recoveries.

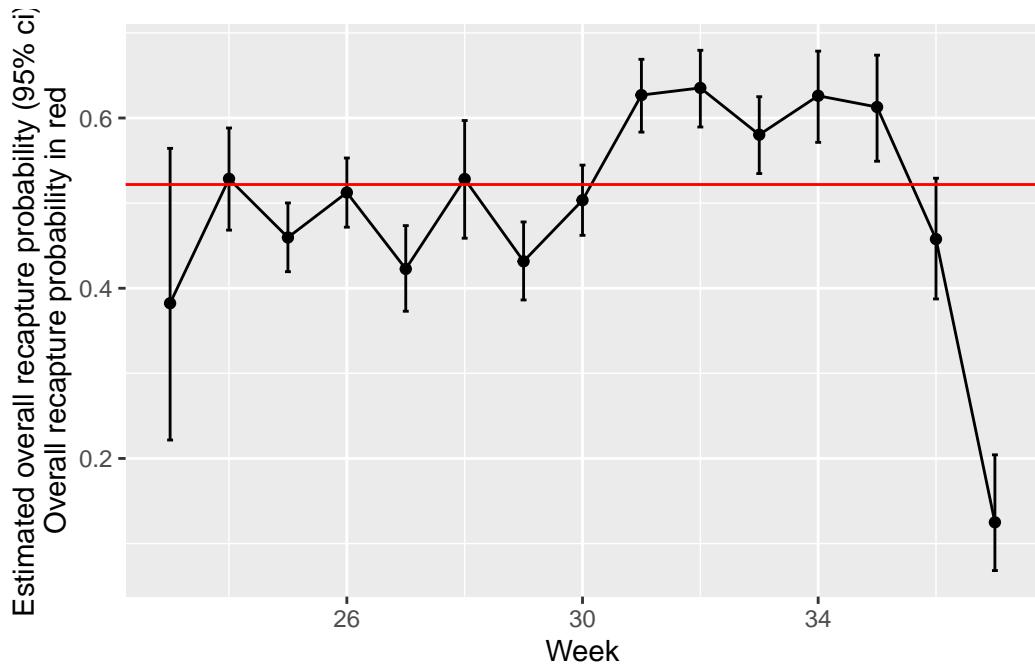


Figure 27: Empirical total recovery probabilities

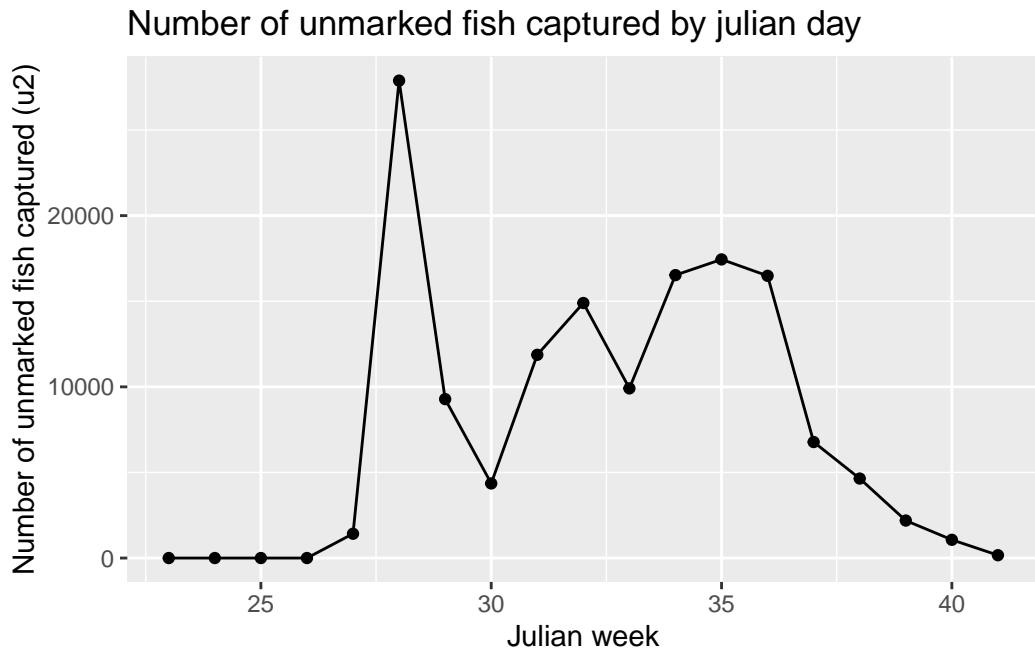


Figure 28: Observed number of unmarked recaptures

8.3.5.2 Fitting the basic BTSPAS non-diagonal model

The *Petersen* package function *BTSPAS_NonDiag_fit()* is a wrapper to the corresponding function in the *BTSPAS* package that takes the (modified for bad events) data file, a model for the mean capture probabilities at the second temporal stratum (default is a common mean), and identification of break points in the underlying spline, and then call the function in the *BTSPAS* package, and formats the returned data structure to match the returning structure for other functions in this package.

If finer control over the fitting process is needed, the *BTSPAS* package should be called directly – please consult the vignettes that come with the *BTSPAS* package.

In this case, no sampling was done in weeks 23 to 27 and so we set the capture probability of 0 on these days:

```
BTSPAS.nondiag.fit1 <- LP_BTSPAS_fit_NonDiag(
  data=data_btspas_nondiag1,
  p_model=~1,
  InitialSeed=34343,
  logitP.fixed=c(23:26), logitP.fixed.values=rep(-10,length(23:26))
)
```

The distribution of the posterior sample for the total number unmarked and total abundance is shown in Figure 29

A plot of the *logit(P)* (the logit of the estimated probability of capture) is shown in Figure 30

A summary of the posterior for each parameter is also available. The estimates of total abundance can be extracted and summarized in a similar fashion as in the other models:

```
BTSPAS.nondiag.est1 <- Petersen::LP_BTSPAS_est (BTSPAS.nondiag.fit1)
BTSPAS.nondiag.est1$summary
```

	N_hat_f	N_hat_rn	N_hat	N_hat_SE	N_hat_conf_level	N_hat_conf_method	
1	NA	NA	336026.4	33118.42	0.95	Posterior	
	N_hat_LCL	N_hat_UCL	p_model	name_model	cond.ll	n parms	nobs
1	286401.8	414635.5	~1	p: ~1	NA	NA	150429

The estimated total abundance from *BTSPAS* is 336,026 (SD 33,118) fish.

Posterior plots with 95% credible intervals

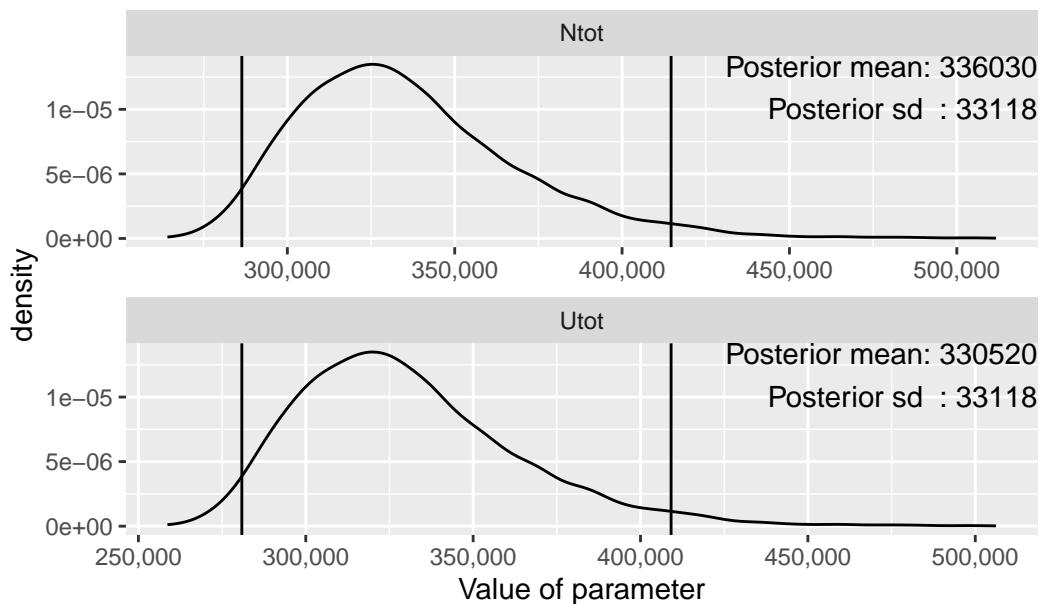


Figure 29: Distribution of posterior samples for total number unmarked and total abundance from BTSPAS diagonal model

Plot of logit($p[i]$) with 95% credible intervals

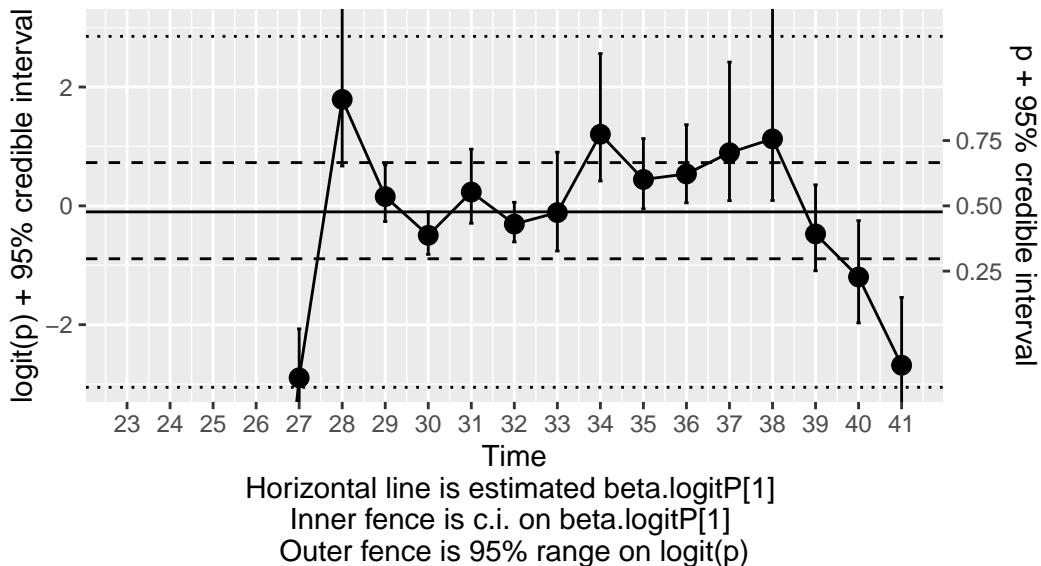


Figure 30: Plot of estimated logit(P) from BTSPAS diagonal model

9 Incomplete or partial stratification

9.1 Introduction

Capture heterogeneity is known to cause bias in estimates of abundance in capture–recapture studies. This heterogeneity is often related to observable fixed characteristics of the animals such as sex. If this information can be observed for every handled animal at both sample occasions, then it is straight-forward to stratify (e.g., by sex) and obtain stratum-specific estimates.

However, it may be difficult to measure the covariate for every fish. For example, it may be difficult to sex all captured fish because morphological differences are slight or because of logistic constraints. In these cases, a subsample of the captured fish at each sample occasion is selected, and additional and often more costly measurements are made, such as sex determination through sacrificing the fish.

The resulting data now consist of two types of marked animals: animals whose value of the stratification variable is unknown, and subsamples at each occasion where the value of the stratification variables are determined.

Premarathna, Schwarz, and Jones (2018) developed methodology for these types of studies. Furthermore, given the relative costs of sampling for a simple capture and for processing the subsample, optimal allocation of effort for a given cost can be determined. They also developed methods to account for additional information (e.g., prior information about the sex ratio) and for supplemental continuous covariates such as length.

These methods were applied to a problem of estimating the size of the walleye population in Mille Lacs Lake, MN and the data are reanalyzed here.

Note that this Petersen package functions for this methods are simple wrappers around the published code from the paper. Furthermore, wrapper are available only for model without continuous covariates in the Petersen package. Contact the above authors for code for more complex cases.

The complete statistical theory is presented in Premarathna, Schwarz, and Jones (2018).

9.2 Sampling protocol

At the first sample occasion, a random sample of size n_1 is captured. Then, a subsample of size n_1^* is selected from n_1 , and the stratum is determined for all animals in the subsample. All captured animals are tagged, usually with a unique tag number, and released back to the population.

Again, some time later, another sample of animals of size n_2 is captured randomly from the population. The animals captured at the second sample occasion contain animals captured

and marked at the first occasion (some of them might be stratified and some might not be stratified), as well as animals not captured at the first occasion. One of the requirements here is that some of the stratified subsamples at the first occasion are recaptured. Also, animals must not be sacrificed to determine stratification membership at the first sample occasion. Again, a subsample of size n_2^* is selected from the captured sample at the second sample occasion including **only** animals not marked at the first occasion. A pictorial view of the sampling protocol is given in Figure 31

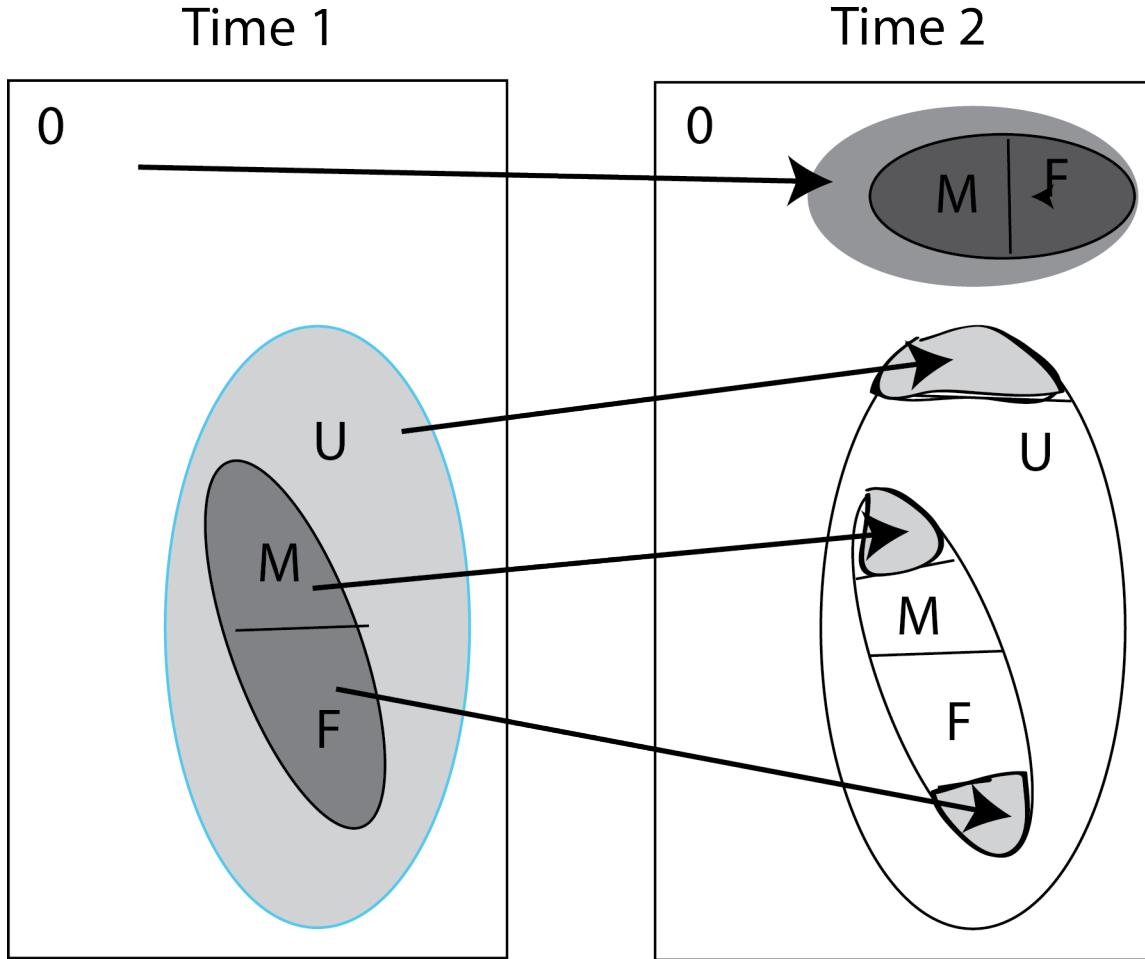


Figure 31: Schematic of sampling protocol for incomplete stratification studies

Note that only unmarked fish are examined in the subsample at both sampling occasions

9.3 Capture histories and parameters

The parameters for this type of study are

- N - population abundance
- p_1, p_2 - capture probabilities at the two sampling occasions
- $\lambda_1, \dots, \lambda_k$ the proportion of the population in the k categories. Must add to 1.
- θ_1, θ_2 - the subsampling proportions at the two occasions.

There are $3k + 4$ possible capture histories. Each history is a two-digit string with entries of 0 (not captured), U (stratification unknown), or a code for the stratification variable (e.g. M, F , etc.).

The possible histories and corresponding probabilities are:

History	Probability
MM	$\lambda_M p_{1M} \theta_1 p_{2M}$
M0	$\lambda_M p_{1M} \theta_1 (1 - p_{2M})$
0M	$\lambda_M (1 - p_{1M}) p_{2M} \theta_2$
...	Similarly for other categories
UU	$\lambda_M p_{1M} (1 - \theta_1) p_{2M} + \lambda_F p_{1F} (1 - \theta_1) p_{2F}$
U0	$\lambda_M p_{1M} (1 - \theta_1) (1 - p_{2M}) + \lambda_F p_{1F} (1 - \theta_1) (1 - p_{2F})$
0U	$\lambda_M (1 - p_{1M}) p_{2M} (1 - \theta_2) + \lambda_F (1 - p_{1F}) p_{2F} (1 - \theta_2)$
00	Everything else (not observable)

Note that histories such as UF are not allowed as only unmarked fish in the second sample are sub-sampled. Also histories such as MF would indicate an error.

Standard numerical likelihood methods are applied in the usual way to estimate the parameters and obtain measures of uncertainty. After estimates are obtained, stratum specific abundance estimates are obtained as $\widehat{N}_k = \widehat{N} \times \widehat{\lambda}_k$ and measures of uncertainty obtained using the delta method.

9.4 Example - Walleye in Milles Lac, MN.

This is the data analyzed in Premarathna, Schwarz, and Jones (2018). There are some slight differences from the data in the published paper because fish that did not have length also measured are removed.

Walleye are captured on the spawning grounds. Almost all the fish can be sexed in the first occasion. All the captured fish are tagged and released and the recapture occurred 3 weeks later using gill-nets. From a sample of fish captured at second occasion that are not tagged, a random sample is selected and sexed.

Here is the summary data:

```
data(data_wae_is_short) # does not include length covariate  
data_wae_is_short
```

	cap_hist	freq
1	OF	237
2	OM	41
3	OU	3058
4	FO	1555
5	FF	32
6	MO	5071
7	MM	40
8	UO	42
9	UU	1

We start by fitting a model that allows for different probabilities of capture by sampling occasion and sex. Notice that we specify the “hidden” stratification variable using `..cat` in the model for p . It is possible to specify models for λ (category proportions) and θ (subsampling fractions) but these are seldom useful (but see an example in the published paper).

Premarathna, Schwarz, and Jones (2018) provided a function to print the full results in a nice format

```
LP_IS_print(wae.res1)
```

Model information:

```
Model Name: p: ~-1 + ..cat:..time; theta: ~-1 + ..time; lambda: ~-1 + ..cat; offsets 0,0  
Neg Log-Likelihood: -71018.68  
Number of Parameters: 8  
AICc value: -142021.4
```

Raw data:

```
$History  
[1] "OF" "OM" "OU" "FO" "FF" "MO" "MM" "UO" "UU"  
  
$counts
```

```
[1] 237 41 3058 1555 32 5071 40 42 1
```

```
$category  
[1] "F" "M"
```

Initial values used for optimization routine:

Initial capture probabilities:

	time1	time2
F	0.0214139	0.01082925
M	0.0214139	0.01082925

Initial category proportions:

lambda_F	lambda_M
0.5	0.5

Initial sub-sample proportions:

theta_1	theta_2
0.99362112	0.08333333

Initial population size for optimization(simple Lincoln Petersen estimator is used) : 314,798

Design matrix and OFFSET vector for capture probabilities:

	Beta...catF...time1	Beta...catM...time1	Beta...catF...time2
p1F	1	0	0
p1M	0	1	0
p2F	0	0	1
p2M	0	0	0

	Beta...catM...time2	OFFSET.vector
p1F	0	0
p1M	0	0
p2F	0	0
p2M	1	0

Design matrix and OFFSET vector for sub-sample proportions (theta):

	Beta...time1	Beta...time2	OFFSET.vector
theta_1	1	0	0
theta_2	0	1	0

```
Design matrix and OFFSET vector for category proportions (lambda):
    ..catF OFFSET.vector
lambda_F      1          0
```

Find MLEs:

MLEs for capture probabilities:

	time1	time2
F	0.01146535	0.020643986
M	0.07658161	0.007930921

MLEs for category proportions:

	lambda_F	lambda_M
	0.674743	0.325257

MLEs for sub-sample proportions:

	theta_1	theta_2
	0.9936211	0.0833333

MLE for Population size : 206,507

MLE for Population size of category F : 139,339

MLE for Population size of category M : 67,168

SE's of the MLEs

SE's of the MLEs of capture probabilities:

	time1	time2
F	0.002009477	0.003569036
M	0.015121929	0.001240106

SE's of the MLEs of the category proportions:

	lambda_F	lambda_M
	0.06017088	0.06017088

SE's of the MLEs of the sub-sample proportions:

	theta_1	theta_2
	0.000969662	0.004785221

SE of the MLE of the population size: 26,475

SE for Population size of category F : 24,172

SE for Population size of category M : 13,234

```

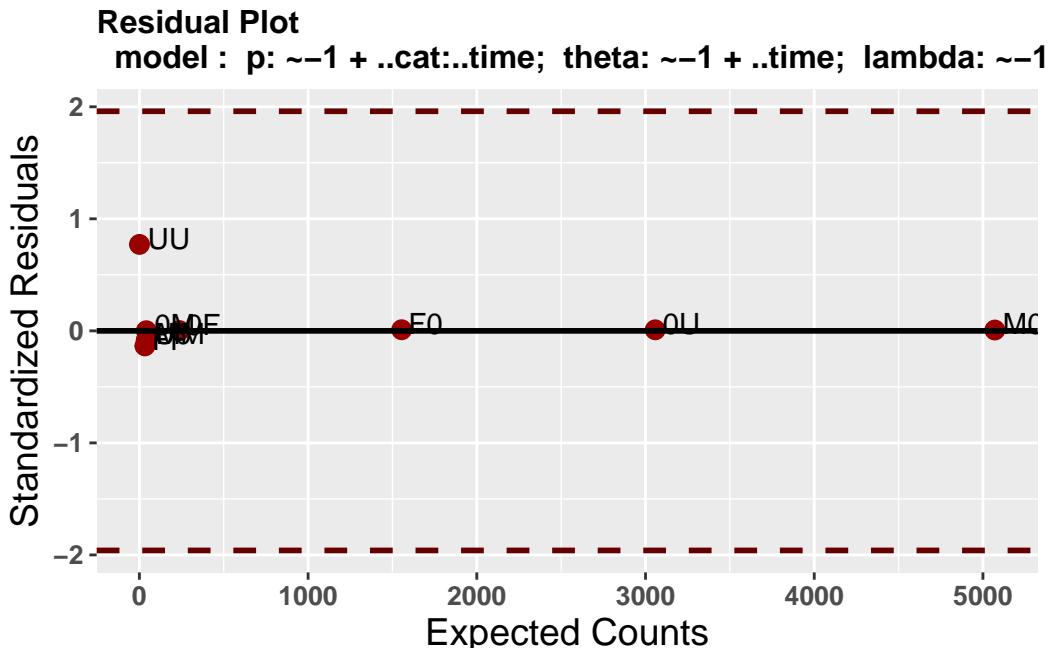
Observed and Expected counts for capture histories for the model p: ~-
1 + ..cat:..time; theta: ~-1 + ..time; lambda: ~-1 + ..cat; offsets 0,0,0,0,0,0,0
   History Observed.Counts Expected.Counts   Residual Standardized.Residuals
1      U0          42    42.5318920 -0.531891968      -0.081566309
2      UU           1    0.4706046  0.529395437       0.771707324
3      F0         1555  1554.6107699  0.389230058       0.009909151
4      M0         5071  5070.4718173  0.528182724       0.007510317
5      FF           32    32.7698643 -0.769864335     -0.134496611
6      MM           40    40.5349888 -0.534988778     -0.084037335
7      OF          237   236.9610300  0.038969959       0.002533033
8      OM           41    40.9922556  0.007744408       0.001209708
9      OU          3058  3057.4874952  0.512504751       0.009338016

```

```

Standardized residual plot for the model p: ~-1 + ..cat:..time; theta: ~-
1 + ..time; lambda: ~-1 + ..cat; offsets 0,0,0,0,0,0,0

```



Estimates of abundance are found in the usual fashion, both for the entire population:

```

wae.est1 <- LP_IS_est(wae.res1, N_hat=~1, conf_level=0.95)
wae.est1$summary

```

```

N_hat_f N_hat_conf_level N_hat_conf_method N_hat_rn      N_hat N_hat_SE

```

```

1      ~1          0.95          logN      All 206506.9 26474.59
      N_hat_LCL N_hat_UCL      p_model  theta_model lambda_mode
1 160623.4 265497.4 ~-1 + ..cat:..time ~-1 + ..time ~-1 + ..cat
                                         name_model
1 p: ~-1 + ..cat:..time; theta: ~-1 + ..time; lambda: ~-1 + ..cat; offsets 0,0,0,0,0,0,0
  cond.ll n parms nobs method
1 71018.68      8 10077 full 11

```

and for the individual strata:

```
wae.est1a <- LP_IS_est(wae.res1, N_hat=~-1+..cat, conf_level=0.95)
wae.est1a$summary
```

```

      N_hat_f N_hat_conf_level N_hat_conf_method N_hat_rn      N_hat N_hat_SE
1  ~-1 + ..cat          0.95          logN      F 139339.1 24172.49
1.1 ~-1 + ..cat          0.95          logN      M 67167.8 13233.86
      N_hat_LCL N_hat_UCL      p_model  theta_model lambda_mode
1   99176.04 195766.78 ~-1 + ..cat:..time ~-1 + ..time ~-1 + ..cat
1.1 45651.13 98825.89 ~-1 + ..cat:..time ~-1 + ..time ~-1 + ..cat
                                         name_model
1  p: ~-1 + ..cat:..time; theta: ~-1 + ..time; lambda: ~-1 + ..cat; offsets 0,0,0,0,0,0,0
1.1 p: ~-1 + ..cat:..time; theta: ~-1 + ..time; lambda: ~-1 + ..cat; offsets 0,0,0,0,0,0,0
  cond.ll n parms nobs method
1  71018.68      8 10077 full 11
1.1 71018.68      8 10077 full 11

```

As with the other functions in this package, we can fit other models such as a pooled-Petersen model:

and compare models using AICc (Table 26)

```
aic.table <- LP_AICc(wae.res1, wae.res2)
```

Table 26: Model comparison in incomplete stratification fits

Model specification	Conditi- onal log- likelihood	# parms	# obs	AICc	Delta	AICcWt
p: ~1 + ..cat:..time; theta: ~1 + ..time; lambda: ~1 + ..cat; offsets 0,0,0,0,0,0,0	71,018.68	8	10,077	142,021.35	-	0.00
p: ~1 + ..time; theta: ~1 + ..time; lambda: ~1 + ..cat; offsets 0,0,0,0,0,0,0	70,786.44	6	10,077	141,560.86	-	460.49

and it is clear that the pooled-Petersen has no support.

Model-averaging is unnecessary given the negligible support for the second fitted model, but again can be done in the usual fashion for the entire population abundance (Table 27):

```
ma.table <- LP_modavg(wae.res1, wae.res2)
```

Table 27: Model averaging in incomplete stratification fits

Modnames	AICcWt	Estimate	SE
p: ~1 + ..cat:..time; theta: ~1 + ..time; lambda: ~1 + ..cat; offsets 0,0,0,0,0,0,0	1.00	206,507	26,475
p: ~1 + ..time; theta: ~1 + ..time; lambda: ~1 + ..cat; offsets 0,0,0,0,0,0,0	0.00	314,672	36,232
Model averaged		206,507	26,475

and for the individual categories (Table 28).

```
ma.table <- LP_modavg(wae.res1, wae.res2, N_hat=~-1+..cat)
```

Table 28: Model averaging in incomplete stratification fits for individual sex strata

Modnames	AICcWt	Estimate	SE
p: ~-1 + ..cat::time; theta: ~-1 + ..time; lambda: ~-1 + ..cat; offsets 0,0,0,0,0,0,0	1.00	139,339	24,172
p: ~-1 + ..time; theta: ~-1 + ..time; lambda: ~-1 + ..cat; offsets 0,0,0,0,0,0,0	0.00	82,275	9,617
Model averaged		139,339	24,172
p: ~-1 + ..cat::time; theta: ~-1 + ..time; lambda: ~-1 + ..cat; offsets 0,0,0,0,0,0,0	1.00	67,168	13,234
p: ~-1 + ..time; theta: ~-1 + ..time; lambda: ~-1 + ..cat; offsets 0,0,0,0,0,0,0	0.00	232,398	26,810
Model averaged		67,168	13,234

9.5 Optimal allocation

In an incomplete-stratified two-sample capture–recapture study, there is a cost to capturing an animal at each sample occasion, a cost to identify the category of the captured animal in the subsamples, and also a fixed cost regardless of the sample size. If there is a fixed amount of funds (C_0) to be used in the study, then the objective is to find the optimal number of animals to capture at both sample occasions and the optimal sizes of the subsamples to be categorized so that the variance of the estimated population abundance ($\text{Var}(\widehat{N})$) is minimized.

The total cost (C) of the study can be considered as a linear function of sample sizes and is given by

$$C = c_f + n_1 c_1 + n_1^* c_1^* + n_2 c_2 + n_2^* c_2^* \leq C_0$$

where

- $n_1^* \leq n_1$
- $n_2^* \leq n_2 - E(n_{UU} + \sum_C n_{cc})$

Numerical optimization methods can be used to find the optimal allocation of n_1 , n_2 , n_1^* , and n_2^* with respect to the linear constraint such that $\text{Var}(\widehat{N})$ is minimized.

The following costs were considered for the analysis of optimal allocation:

- $c_f = 0$,
- $c_1 = 4$,
- $c_1^* = 0.4$,
- $c_2 = 6$,

- $c_2^* = 0.4$, and
- $C_0 = 90,000$.

These costs are the time in minutes for the operation on a fish. A value of $C_0 = 90,000$ minutes (i.e., 1,500 h) available for this study.

Optimal allocation suitable guesstimates for the parameters using previous studies or the researcher's experience. We used the following guesstimates for the parameters:

- $N = 209,000$,
- $\lambda_M = 0.33$,
- $r_1 = 6.5$, and
- $r_2 = 0.4$

where r_1 is the guesstimate of the ratio of p_{1M} / p_{1F} , and r_2 is the guesstimate of the ratio of p_{2M} / p_{2F} .

It is difficult to give guesstimates for capture probabilities for each category at each sample occasion. However, in practice, we can give a ratio of the capture probabilities at each sample occasion. For example, if detection probability of males is half that of females at the first sample occasion, then the ratio r_1 is 0.5. The guesstimates of the ratios r_1 and r_2 given above are calculated using the estimates for capture probabilities see previously.

Optimal allocations of sample sizes and subsample sizes produced by numerical methods for the given costs are $n_1 = 8,929$, $n_1^* = 8,908$, $n_2 = 8,359$, and $n_2^* = 1,412$. At the optimal allocation, $SE(\widehat{N}) = 13,657$ which is 50% lower than the SE obtained using the current allocation.

Different solutions are available for n_1 , n_2 , n_1^* , and n_2^* at optimal allocation. Conditional contour plots were used to see these different solutions. A conditional contour plot in this situation is a contour plot for the standard error of \widehat{N} where two values of n_1 , n_2 , n_1^* , or n_2^* are fixed at the optimal values. A conditional contour plot for standard error of \widehat{N} when n_1^* and n_2^* are fixed at the optimal values and when n_1 and n_2 are fixed at the optimal values are given in Figure 32. These contour plots show that many solutions are possible for optimal allocation.

Full R code is available from the paper authors.

9.6 Additional details

Premarathna, Schwarz, and Jones (2018) also provide details on

- determining the sample size for detecting differential catchabilities (power analysis);
- determining the approximate standard error to be expected under proposed sample sizes (sample size analysis),
- performing goodness-of-fit test for the chosen model,
- using individual covariates (such a length) and an conditional likelihood approach, and
- using a Bayesian analysis to incorporate outside information (such as the sex ratio).

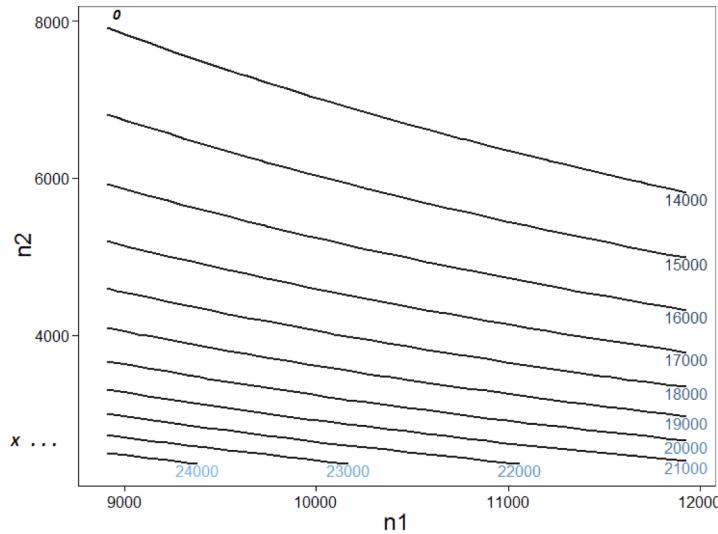


Figure 32: Conditional contour plot for SE of estimated abundance. o=optimal allocation; x=current allocation (outside plot boundaries)

Please consult their paper for more details.

We need to consider the issue of nonidentifiability in model fitting. All the parameters can be estimated in two-sample capture–recapture studies similar to Lincoln–Peterson (Williams et al., 2002) model fitting. However, we would not be able to estimate some of the parameters, for example, if no females were observed. There is no nonidentifiable issue on model fitting with walleye data because both males and females were observed. However, there can be a nonidentifiability issue with the population abundance parameter when modeling with individual heterogeneity, as described by Link (2003).

10 Double tagging studies to account for tag loss or non-reporting

10.1 Introduction

Tag loss is one of the few violations of assumptions that can be most easily accounted for by a modification of the sampling protocol.

Tag loss leads to a positive bias in estimates of population size because fewer tagged fish are “apparently” recovered, i.e., fish that lost tags look as if they are untagged. In order to estimate and to adjust for tag loss, some (or all) of the fish released must be double tagged.

The analysis of double tagging studies was described by Gulland (1963), Beverton and Holt (1957), Seber and Felton (1981) and Hyun et al (2012). Seber (1982) and Weatherall (1982) have a nice summary of dealing with tag loss and *ad hoc* adjustments to estimates of abundance. It is generally assumed that a fish that loses its tags is indistinguishable from fish that were not tagged originally.

This double tagging generally takes two forms:

- two identical or two different tags are applied. Either or both could be lost, and the retention probability could be different for each tag.
- a permanent mark (such as a fin clip or a permanent dye injection or a ‘genetic tag’) is applied. Fish that are recovered with the permanent mark but without a tag are known to have lost a tag, but the individual cannot be determined.

When double tagging fish, tag 1 of the double tag should be the same type and placed in the same location on the fish as singly tagged fish. If tags have tag numbers inscribed on the fish, you can identify from a recovery of a fish with a single tag, if it was singly or doubly tagged. Hyun et al (2012) used a different colored tags for the double tagged fish so that a fish with a single tag captured at the second event can be classified as either initially singly or doubly tagged.

The second tag can be the same tag type placed in a different location, a different tag (e.g. a PIT tag), or a permanent batch mark.

Models with a permanent double-tag are simplification of the models with tags than can be lost where the tag retention probability is set to 1. Some care is needed if a batch mark is used because you may also need to know the stratum of release. If the stratum of release is based on fish characteristics, then these can usually be measured at the second event, but if the stratum of release is based on geographic or temporal strata, then the batch mark may not be sufficient to identify the stratum of release.

Tag loss has been traditionally divided into two types (Beverton and Holt, 1957):

- Type I losses are losses that occur immediately after tagging, e.g. immediate tag shedding. This type of tag loss reduces the effective number of tags initially put out.
- Type II losses are those that happen steadily and gradually over an extended period of time following release of the tagged fish.

In a simple Petersen study, the two types of tag loss cannot be distinguished and the total tag loss between the two sampling occasions is estimated. If the exact time at liberty of tags is known, then Weatherall (1982) reviews how to estimate the two types of losses. If multiple sampling occasions take place (i.e., a Jolly-Seber study), Arnason and Mills (1981), and McDonald, Amstrup and Manly (2003) examine the biases that can be introduced, and Cowen and Schwarz (2006) develops methodology to correct for tag loss.

The *Petersen* package can deal with three types of double tagging studies:

- Two indistinguishable double tags are used so that if a fish is recovered with only one of the double tags, you do not know which tag was lost. This type of studies often occurs when fish are not directly handled so tag numbers inscribed on the tag cannot be read. Similarly, the tags must placed close together so that it is difficult to distinguish. If for example, double tags were placed on the left and right of a fish, or front and back of a fish, then this may an example of the second case.
- Two distinguishable double tags are used so that if a fish is recovered with only one of the double tags, you do know which of the double tags were lost. For example, every tag could have a unique tag number; or applied to the left/right side of a fish
- A permanent batch mark (e.g., fin clip) is applied that is assumed cannot be lost.

10.2 Capture histories

The **apparent** capture history of a fish is again denoted using ω . It is important to remember that the **apparent** capture-history may not reflect the underlying actual capture history. For example, a tagged fish could lose all its tags before being recaptured at the second event. This capture looks like the capture of an untagged fish, but actually is a double counting of a fish with a different **apparent** capture history.

The **apparent** capture history is expanded to also reflect the observed tagging status. A two-digit code is code for each sample event, with the first digit (0 or 1) indicating if tag 1 is present on the fish and the second value (0 or 1) indicating if tag 2 is present on the fish.

Each component of the tag-history vector is generalized to represent the status of each tag at each capture occasion Cowen and Schwarz (2006). For example, here are the possible history vectors when double tagging is performed:

For fish that received a single tag, possible histories are:

- (10 10): Fish single tagged with tag type A at occasion 1 and recaptured at sampling occasion 2.
- (10 00): Fish single tagged with tag type A at occasion 1 that are “not seen” at occasion 2. Note that this consists of fish which have retained their tag and were not captured at occasion 2 and fish that lost their tag and were recaptured at occasion 2. The latter are not recognizable has having being previously tagged.

For fish that received two indistinguishable double tags, possible histories are:

- (11 00): Fish double tagged but not recaptured with at least one of its tags present.
- (11 1X): One of the double tags was present, but it is not known which
- (11 11): Fish double tagged and recovered with both tags present.

For fish that received two distinguishable double tags, the history (11,1x) is divided into

- (11 10): Fish double tagged with tag types B at occasion 1, and recaptured only with the double tag in location 1
- (11 01): Similar to above, but now recaptured with double tag in location 2 only present

For fish that received a permanent second tag, possible histories are:

- (1P 00): Fish double tagged, but not recaptured with at least one of its tags present.
- (1P 0P): Fish double tagged, but one tag (non permanent) has been lost
- (1P 1P): Fish double tagged, and recovered with both tags present.

Finally, we have fish that apparently are recaptured for the first time at the second event.

- {00 10}: Fish that apparently are seen for the first time at the second sampling occasion. This includes fish captured for the first time at the second occasion, and **also fish that were tagged at occasion 1, lost their tag(s) between the two sampling occasions, and were recaptured at occasion 2.**

Now, in the case with two distinguishable tags

$$n_1 = n_{1010} + n_{1000} + n_{1111} + n_{1101} + n_{1110} + n_{1100}$$

$$n_2 = n_{1010} + n_{1101} + n_{1110} + n_{1111} + n_{0010}$$

but a complete count of recaptured fish cannot be made because some fish lost all their tags and were captured and so have an apparent history of (00 10).

Similar equations can be derived for the other two cases.

The number of each fish in the study with the observed capture history is the frequency count and is denoted as n_ω .

The additional parameters to be introduced into the model are the tag retention probabilities at the two locations (ρ_1 , and ρ_2) and the proportion of fish that are single tagged at the first sampling occasion, p_{ST} which is assumed to be known given the ratio of single and double tagging done at the first sampling occasion.

A complete likelihood analysis based on capture histories is quite complex because of fish that loose all their tags and are recaptured. Hyun et al (2012) developed a likelihood-approach based on the summary statistics, but this does not easily allow for stratification, or if the capture and/or tagloss retention probabilities depend on covariates. In Appendix C we present a conditional likelihood approach similar to the approach used in the conditional likelihood Petersen estimator.

In this conditional likelihood approach we condition on fish captured at the second event and ignore histories of fish tagged and never seen again. This is a sensible approach because we cannot estimate the capture probability at the second event because the **apparent** untagged animals captured at the second occasion are mixture of truly untagged fish and fish that were

captured at the first sampling occasion and lost all their tags. Then a HT-type estimator is used to estimate the abundance using the number of fish captured at the first occasion. Consequently, you can only develop a model for the catchability at the first sampling event (e.g., depends on discrete or continuous covariates).

- try and double tag as many fish as possible, esp if tag retention is low
- one of the double tags should match the single tag other wise you have 3 tag types and not much can be done.
- hypothesis test for tagging of the form “no tag loss” are silly. If any tag loss is observed, then the hypothesis is obviously rejected. If tag loss is very low, then you can fit a simple Petersen study and live with the small bias.

10.3 Example - Kookanee on Metolius River - two tags that cannot be distinguished

This is the data analyzed in Hyun et al (2012).

In August and September 2007, the period just before the spawning run, adult kokanee were collected by beach seining in the upper arm of the lake near the confluence with the Metolius River. Fish were tagged with nonpermanent, plastic T-bar anchor tags and then were released back into the lake. Randomly selected fish received single tags of one color, while the other fish received two tags of a second color (i.e., the double tags were identical in color). In late September through October, spawning ground surveys were conducted by 2–3 people walking abreast in a downstream direction (or floating, in sections where the water depth and flow were too great to allow walking). Instead of being physically recaptured, the fish were resighted as they swam freely in the clear, relatively shallow water within the spawning areas of the river. The total number of fish observed with or without a tag (or tags) was recorded for each section, and information on the number and color of tags for each marked fish was also noted.

Notice that tags where only a single tag is applied to a fish are a different color than tags where double tags are applied so that a fish that is subsequently detected with a single tag can be identified if it originally had a one or two tags applied.

The data were extracted from Hyun et al (2012) and converted to capture-histories.

```
data(data_kokanee_tagloss)
data_kokanee_tagloss[,1:2]
```

	cap_hist	freq
1	1000	2589
2	1010	218
3	1111	35
4	111X	24

```

5      1100   432
6      0010 11167

```

Because fish were not handled, it is not possible to know which of the double tags were lost, (i.e., we cannot distinguish between histories 1110 and 1101 and these are pooled into capture history $111X$). Only models with equal retention probabilities for the two tags can be fit.

```

kok.res1 <- LP_TL_fit(data_kokanee_tagloss,
                      dt_type="notD",
                      p_model=~1,
                      rho_model=~1)
kok.res1$summary

  p_model rho_model      name_model cond.ll n parms   nobs      method
1       ~1        ~1 p: ~1; rho: ~1 -1488.626      2 11420 TL cond ll

```

Estimates of abundance are:

```

kok.est1 <- LP_TL_est(kok.res1, N_hat=~1, conf_level=0.95)
kok.est1$summary

  N_hat_f    N_hat_rn    N_hat N_hat_SE N_hat_conf_level N_hat_conf_method
1       ~1 (Intercept) 103035.9 9099.521          0.95           logN
  N_hat_LCL N_hat_UCL p_model rho_model      name_model cond.ll n parms
1 86659.33 122507.2      ~1        ~1 p1: ~1; rho: ~1 -1488.626      2
  nobs      method
1 14465 LP_TS CondLik

```

The conditional likelihood estimate is 103 (SE 9) thousand fish compared to the estimates of 103 (SE 9) thousand fish obtained by Hyatt et al (2012).

Estimates of the tag retention probability is

```

kok.est1$detail$data.expand[1:2,c(..tag","rho","rho.se")]

```

	..tag	rho	rho.se
2	1	0.7266325	0.05117941
2.1	2	0.7266325	0.05117941

These estimates match those from Hyatt et al (2012) of tag retention of $1 - .27 = .73$ (SE .05).

In this example, we conditioned on being captured at time 2 so avoid double counting fish that were tagged, lost all tags, and then were recaptured.

It is estimated that 85 single tagged fish lost their tag and were recaptured at the second event, and 4 double tagged fish lost all their tags and were recaptured at the second event. If these are added to the 277 **OBSERVED** recaptures at the second event, it is estimated that the total number of recaptures (observed and unobserved) is 366 fish. The regular-Petersen estimator using the observed values of $n_1 = 3298$ and $n_2 = 1.1444 \times 10^4$ is $\widehat{N}_{Petersen} = 103,036$ fish which matches very closely (as it must) to the estimates from the model. Of course, the estimated SE from the Petersen estimator using the estimated number of recaptures will be too small because it does not account for the uncertainty in estimating the expected number of recaptures.

10.4 Example - Simulated data - two tags that can be distinguished

Simulated data are used for this example. **Tried but could not find numerical example in literature**

The observed capture history data is:

```
data(data_sim_tagloss_twoD)
data_sim_tagloss_twoD[,-(1:2)]
```

	cap_hist	freq
1	0010	879
2	1000	225
3	1010	14
4	1100	666
5	1101	21
6	1110	7
7	1111	37

Notice that we now have histories (11,01) and (11,10) compared to the previous example where these were “combined” into history (11,1X).

We can not fit models with and without equal tag retention probabilities and compare them in the usual fashion.

10.4.1 Model with equal tag retention probabilities

First, the model with equal retention probabilities across the two tag locations:

```
twoD.res1 <- LP_TL_fit(data_sim_tagloss_twoD,
                         dt_type="twoD",
                         p_model=~1,
                         rho_model=~1)
twoD.est1 <- LP_TL_est(twoD.res1, N_hat=~1, conf_level=0.95)
twoD.est1$summary
```

	N_hat_f	N_hat_rn	N_hat	N_hat_SE	N_hat_conf_level	N_hat_conf_method
1	~1	(Intercept)	10267.11	1214.854	0.95	logN
	N_hat_LCL	N_hat_UCL	p_model	rho_model	name_model	cond.ll n parms nobs
1	8141.981	12946.92	~1	~1	p1: ~1; rho: ~1	-373.7071 2 1849
	method					
1	LP_TS	CondLik				

Estimates of the tag retention probability is

```
twoD.est1$detail$data.expand[1:2,c(..tag,"rho","rho.se")]
```

	..tag	rho	rho.se
1	1	0.7213776	0.04992898
1.1	2	0.7213776	0.04992898

10.4.2 Model with unequal tag retention probabilities

Second, a model with unequal retention probabilities across the two tag locations is fit

```
twoD.res2 <- LP_TL_fit(data_sim_tagloss_twoD,
                         dt_type="twoD",
                         p_model=~1,
                         rho_model=~1+..tag)
twoD.est2 <- LP_TL_est(twoD.res2, N_hat=~1, conf_level=0.95)
twoD.est2$summary
```

```

N_hat_f      N_hat_rn   N_hat N_hat_SE N_hat_conf_level N_hat_conf_method
1           ~1 (Intercept) 10196.2 1194.993                  0.95          logN
N_hat_LCL N_hat_UCL p_model    rho_model             name_model cond.ll
1 8103.591 12829.18     ~1 ~-1 + ..tag p1: ~1; rho: ~-1 + ..tag -369.8691
n parms nobs       method
1        3 1849 LP_TS CondLik

```

Note the use of the `..tag` variable in the formula for `rho` indicating that tag location is a factor that influences the retention probability.

Estimates of the different tag retention probabilities are

```
twoD.est2$detail$data.expand[1:2,c(..tag,"rho","rho.se")]
```

```

..tag      rho      rho.se
1         1 0.6363982 0.06085315
1.1       2 0.8409085 0.05514061

```

10.4.3 Comparing models and model averaged estimates of abundance

We compare the two models in the usual way (Table 29) and obtain the model averaged estimates of abundance (Table 30).

```
sim.twoD.aictab <- LP_AICc(
  twoD.res1,
  twoD.res2
)
```

Table 29: Comparison of models fit to the simulated tag loss dataset with distinguishable table

Model specification	Condi- tional log- likelihood	# parms	# obs	AICc	Delta	AICcWt
p: ~1; rho: ~-1 + ..tag	-369.8691	3	958	745.76	0.00	0.94
p: ~1; rho: ~1	-373.7071	2	958	751.43	5.66	0.06

Most weight is given to the model with different tag retention probabilities.

```
# extract the estimates of the abundance
sim.twoD.ma.N_hat<- LP_modavg(
  twoD.res1,
  twoD.res2, N_hat=~1)
```

Table 30: Model averaged estimates of abundance for tag loss model with distinguishable tags

Modnames	AICcWt	Estimate	SE
p1: ~1; rho: ~1 + ..tag	0.94	10,196	1,195
p1: ~1; rho: ~1	0.06	10,267	1,215
Model averaged		10,200	1,196

10.5 Example - Simulated data - a permanent second tag used

Simulated data are used for this example. **Tried but could not find numerical example in literature**

The observed capture history data is:

```
data(data_sim_tagloss_t2perm)
data_sim_tagloss_t2perm[,-(1:2)]
```

cap_hist	freq	
1	0010	923
2	1000	254
3	1010	17
4	1P00	668
5	1P0P	26
6	1P1P	59

We can now fit models but the model for the tag retention probability cannot depend on the tag number since the permanent tag has retention probability of 1.

```
t2perm.res1 <- LP_TL_fit(data_sim_tagloss_t2perm,
                           dt_type="t2perm",
                           p_model=~1,
                           rho_model=~1)
t2perm.est1 <- LP_TL_est(t2perm.res1, N_hat=~1, conf_level=0.95)
t2perm.est1$summary
```

```

N_hat_f      N_hat_rn      N_hat N_hat_SE N_hat_conf_level N_hat_conf_method
1 ~1 (Intercept) 9425.522  937.881           0.95          logN
N_hat_LCL N_hat_UCL p_model rho_model      name_model cond.ll n parms nobs
1 7755.452 11455.23 ~1 ~1 p1: ~1; rho: ~1 -430.7471 2 1947
method
1 LP_TS CondLik

```

Estimates of the probability of capture at the first event, and the tag retention probabilities are

```
t2perm.est1$detail$data[1, c('p1', "p1.se")]
```

```

p1      p1.se
1 0.1086412 0.01032415

```

```
t2perm.est1$detail$data.expand[1:2,c(..tag","rho","rho.se")]
```

```

..tag      rho      rho.se
1       1 0.6824881 0.04923361
1.1     2 1.0000000 0.00000000

```

10.6 Example - Northern Pike - tag loss

In Section 6.2, we analyzed the Northern Pike data assuming that tag loss was negligible. In this example, we will estimate the tag retention probability and show that the conditional likelihood tagloss models can include more complex effects.

Refer to Section 6.2 for information on the sampling design.

We load the data in the usual fashion:

```

data(data_NorthernPike_tagloss)
data_NorthernPike_tagloss[1:5,]

```

	cap_hist	Sex	Length	freq
1	0010	m	23.20	1
2	0010	f	28.89	1
3	0010	m	25.20	1
4	0010	m	22.20	1
5	0010	m	25.00	1

Fish with histories 1110 and 1101 are pooled into capture history $111X$ and only models with equal retention probabilities for the two tags can be fit.

We start with a basic model, where capture probabilities and tag-retention probabilities don't vary over the fish. The estimate of abundance is:

```
nop.TL.res1 <- LP_TL_fit(data_NorthernPike_tagloss,
                           dt_type="notD",
                           p_model=~1,
                           rho_model=~1)
nop.TL.est1 <- LP_TL_est(nop.TL.res1, N_hat=~1, conf_level=0.95)
nop.TL.est1$summary
```

	N_hat_f	N_hat_rn	N_hat	N_hat_SE	N_hat_conf_level	N_hat_conf_method
1	~ 1	(Intercept)	49516.75	3711.682	0.95	logN
	N_hat_LCL	N_hat_UCL	p_model	rho_model	name_model	cond.ll n parms nobs
1	42751.13	57353.06	~ 1	~ 1	p1: ~ 1 ; rho: ~ 1	-482.3847 2 7805
	method					
1	LP_TS	CondLik				

Estimates of the tag retention probability is

```
nop.TL.est1$detail$data.expand[1:2,c("../tag","rho","rho.se")]
```

	..tag	rho	rho.se
1	1	0.9805195	0.007951377
1.1	2	0.9805195	0.007951377

So estimated tag retention was over 98%, and that is why tag loss was previously ignored.

We now fit a model stratified by sex and the estimate of overall abundance is now:

```
nop.TL.res2 <- LP_TL_fit(data_NorthernPike_tagloss,
                           dt_type="notD",
                           p_model=~Sex,
                           rho_model=~1)
nop.TL.est2 <- LP_TL_est(nop.TL.res2, N_hat=~1, conf_level=0.95)
nop.TL.est2$summary
```

```

N_hat_f      N_hat_rn      N_hat N_hat_SE N_hat_conf_level N_hat_conf_method
1      ~1 (Intercept) 49363.51 3699.588          0.95      logN
N_hat_LCL N_hat_UCL p_model rho_model      name_model cond.ll n parms
1 42619.86 57174.19 ~Sex      ~1 p1: ~Sex; rho: ~1 -482.0723      3
nobs      method
1 7805 LP_TS CondLik

```

Finally, we allow the tag retention probabilities to also vary by sex and the estimated overall abundance is:

```

nop.TL.res3 <- LP_TL_fit(data_NorthernPike_tagloss,
                           dt_type="notD",
                           p_model=~Sex,
                           rho_model=~Sex)
nop.TL.est3 <- LP_TL_est(nop.TL.res3, N_hat=~1, conf_level=0.95)
nop.TL.est3$summary

```

```

N_hat_f      N_hat_rn      N_hat N_hat_SE N_hat_conf_level N_hat_conf_method
1      ~1 (Intercept) 49363.12 3699.555          0.95      logN
N_hat_LCL N_hat_UCL p_model rho_model      name_model cond.ll n parms
1 42619.53 57173.73 ~Sex      ~Sex p1: ~Sex; rho: ~Sex -482.016      4
nobs      method
1 7805 LP_TS CondLik

```

Estimates of the tag retention probability for the two sexes are

```

firstm <- match("m", nop.TL.est3$detail$data.expand$Sex)
firstf <- match("f", nop.TL.est3$detail$data.expand$Sex)
nop.TL.est3$detail$data.expand[sort(c(firstm, firstm+1, firstf, firstf+1)), c("Sex", ".tag", ".rho")]

```

	Sex	.tag	rho	rho.se
1	m	1	0.9774436	0.013019620
1.1	m	2	0.9774436	0.013019620
2	f	1	0.9828572	0.009895976
2.1	f	2	0.9828572	0.009895976

The tag retention probabilities are very similar across the two sexes.

The usual model averaging can be done as shown in Table 31.

```

nop.TL.aictab <- Petersen::LP_AICc (
  nop.TL.res1,
  nop.TL.res2,
  nop.TL.res3
)

```

Table 31: Model comparison of models fit to Northern Pike double tagging data

Model specification	Condi-tional log-likelihood	# parms	# obs	AICc	Delta	AICcWt
p: ~1; rho: ~1	-482.3847	2	1,134	968.78	0.00	0.59
p: ~Sex; rho: ~1	-482.0723	3	1,134	970.17	1.39	0.30
p: ~Sex; rho: ~Sex	-482.0160	4	1,134	972.07	3.29	0.11

Most of the model weight is on the simplest model. The model averaged overall abundance estimate are (Table 32)

```

nop.TL.modavg <- Petersen::LP_modavg (
  nop.TL.res1,
  nop.TL.res2,
  nop.TL.res3
)

```

Table 32: Model average estimate of overall abundance from models fit to Northern Pike double tagging data

N_hat_f	N_hat_rn	Modnames	AICcWt	Estimate	SE
~1	(Intercept)	p1: ~1; rho: ~1	0.59	49,517	3,712
~1	(Intercept)	p1: ~Sex; rho: ~1	0.30	49,364	3,700
~1	(Intercept)	p1: ~Sex; rho: ~Sex	0.12	49,363	3,700
Model averaged				49,454	3,707

10.7 Example - Simulated data - non-reporting of tags and reward tags

One assumption of the Petersen estimator is that all recovered tags are reported. However, some studies relies on returns from anglers or other citizens, and not all tags may be reported. To estimate the reporting probability, a second type of tag (the “reward” tag) is applied that often has a monetary incentive to return. The monetary incentive should be large enough that the reporting probability for these tags is 100%.

These types of studies can also be analyzed using the tagloss models. The reward tag is treated as a permanent tag.

Simulated data are used for this example. The observed capture history data is:

```
data(data_sim_reward)
data_sim_reward
```

	cap_hist	freq
1	0010	930
2	OP00	207
3	OPOP	23
4	1000	712
5	1010	56

In this example, about 1500 non-reward tags were applied, and about 230 reward tags were applied. No fish has both types of tags. The reporting probability is estimated by the ratio of the tags returned

```
# estimate return probability and empirical SE
t1.applied <- sum(data_sim_reward$freq[ data_sim_reward$cap_hist %in% c("1000","1010")])
t1.returned <- sum(data_sim_reward$freq[ data_sim_reward$cap_hist %in% c("OPOP")])
t2.applied <- sum(data_sim_reward$freq[ data_sim_reward$cap_hist %in% c("OP00")])
t2.returned <- sum(data_sim_reward$freq[ data_sim_reward$cap_hist %in% c("OP00" )])

rr    <- (t1.returned/t1.applied) / (t2.returned/t2.applied)
rr.se <- sd( rbeta(10000, t1.returned, t1.applied-t1.returned)/ rbeta(10000, t2.returned, t2.applied))
```

and is estimated as 0.729 (SE 0.189).

We can now fit the permanent tag models as seen previously and obtain estimates of abundance:

```

reward.res1 <- LP_TL_fit(data_sim_reward,
                           dt_type="t2perm",
                           p_model=~1,
                           rho_model=~1)
reward.est1 <- LP_TL_est(reward.res1, N_hat=~1, conf_level=0.95)
reward.est1$summary

```

```

N_hat_f      N_hat_rn      N_hat N_hat_SE N_hat_conf_level N_hat_conf_method
1           ~1 (Intercept) 10090.01 2101.777                  0.95          logN
N_hat_LCL N_hat_UCL p_model rho_model      name_model cond.ll n parms nobs
1   6707.858  15177.46     ~1           ~1 p1: ~1; rho: ~1 -324.7077      2 1928
method
1 LP_TS CondLik

```

Estimates of the probability of capture at the first event, and the tag reporting probability are

```
reward.est1$detail$data[1, c('p1',"p1.se")]
```

```

p1      p1.se
1 0.09890973 0.02038768

```

```
reward.est1$detail$data.expand[1:2,c(..tag","rho","rho.se")]
```

```

..tag      rho      rho.se
1         1 0.7291672 0.1805853
1.1       2 1.0000000 0.0000000

```

10.8 Planning tag loss studies

As a general rule, unless you have good information on tag retention probabilities, it is better to assume the worst and plan studies under the pessimistic assumption that tag loss will occur. The primary question is then how many fish should be double tagged?

This is a trade off because it is more costly to actually double tag fish, both when applying tags, but also in recording tag numbers, etc. There are additional tagging costs etc.

The easiest way to investigate the optimal sampling design is via simulation.

10.8.1 Example of deciding between proportion of double tagging

Suppose that you have \$15,000 for tagging operations and that double tagging a fish costs 2x as much as single tagging a fish in supplies (more tags) and additional labor. We can simulate a tagging data set with various parameters, and then compare the precision (SE) as a function of the number of double tagged fish.

We start by making some assumptions about the

- abundance say, 10,000 fish.
- tag retention probabilities (say 0.80 for all fish)
- effort at the second sample, say 1000 fish can be examined.

We first generate the scenarios that we wish to compare

```
scenarios <- expand.grid(N=10000,
                           rho1=0.80,
                           rho2=0.80,
                           n2=1000,
                           n1.dt=seq(50, 500, 50), # number of double tags
                           n.sim=10)               # simulations per scenario
scenarios$n1.st <- 1500 - scenarios$n1.dt*3   # how many single tags can be applied
scenarios$n1 <- scenarios$n1.st + scenarios$n1.dt
scenarios$pST<- scenarios$n1.st/ scenarios$n1
scenarios$index <- 1:nrow(scenarios)
scenarios
```

	N	rho1	rho2	n2	n1.dt	n.sim	n1.st	n1	pST	index
1	10000	0.8	0.8	1000	50	10	1350	1400	0.9642857	1
2	10000	0.8	0.8	1000	100	10	1200	1300	0.9230769	2
3	10000	0.8	0.8	1000	150	10	1050	1200	0.8750000	3
4	10000	0.8	0.8	1000	200	10	900	1100	0.8181818	4
5	10000	0.8	0.8	1000	250	10	750	1000	0.7500000	5
6	10000	0.8	0.8	1000	300	10	600	900	0.6666667	6
7	10000	0.8	0.8	1000	350	10	450	800	0.5625000	7
8	10000	0.8	0.8	1000	400	10	300	700	0.4285714	8
9	10000	0.8	0.8	1000	450	10	150	600	0.2500000	9
10	10000	0.8	0.8	1000	500	10	0	500	0.0000000	10

As more fish are double tagged, the total number of tagged fish declines.

Now for each scenario, we simulate some data, analyze it using the *notD* tag loss model and plot the estimated precision vs number of double tags (Figure 33)

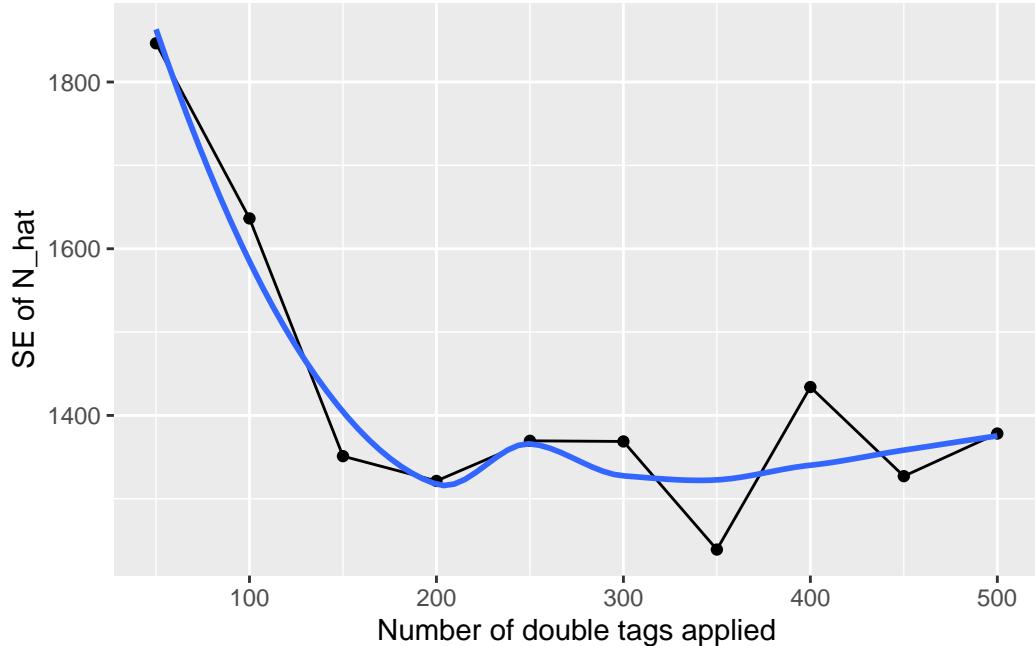


Figure 33: SE of \hat{N} vs. number of double tagged fish

The wiggliness of the line is due to the small number of simulations run for each scenario.

We see that the SE of \hat{N} declines rapidly as you increase the number of double tagged fish to about 200, and then slowly increases because the gain in precision from estimating the tag retention probabilities better with more double tagged fish is offset by the loss in precision by tagging fewer fish.

11 Multiple marks

11.1 Introduction

In some cases, multiple marks can be applied to an animal, but not all marks can be read for an encountered animal. The most common example of this are capture-recapture studies using natural marks found on the left or right side of an animals and “captures” are photographs.

Unfortunately, use of photo-identification records in mark–recapture analyses is not entirely without its problems. For instance, when natural markings are bilaterally asymmetrical, matching photographs to individuals can be difficult when investigators are routinely able to photograph only one side of an animal. The typical approach in the literature when confronted with this situation is to conduct separate analyses with left-sided and right-sided photographs

and compare the results. The same animal may be “captured” more than once in a sampling event, but the two records, from the right or left side of the animal, cannot be matched.

Bonner and Holmberg (2013) and McClintock et al. (2013) simultaneously presented the theory for these analyses and McClintock (2015) created an *R* package (*multimark*) for the analysis of these experiments.

Much of the following is taken from these papers as applied to a two-sample (Petersen-type) experiment.

11.2 Theory

Consider a “typical” Petersen capture–recapture experiment where sampling is conducted over 2 sampling occasions and the identity of each animal is known with certainty when it is observed based on applied tags or marks. As seen previously, there are four possible encounter histories, three of which are directly observable:

- 01 (encountered on the second occasion but not the first),
- 10 (encountered on the first occasion but not the second), and
- 11 (encountered on both occasions),
- 00 (not encountered on either sampling event)- not observable.

In contrast to the preceding scenario, now consider the situation in which individuals are encountered via photographs and marks are bilaterally asymmetrical.

We will consider two photo-sampling scenarios, where on each sampling event, a

- 0 represents a non-encounter,
- L represents an encounter on the left side only as L,
- R represents an encounter on right-sided only, and
- B represent an encounter on both sides. Depending on the design of camera surveys, *B* encounters may be recorded from a single image (where both sides of an individual are visible), a pair of synchronized images, or multiple images (assuming both sides of an individual are known).

One-sided analyses are common for bilateral encounter data when researchers are unable to match opposite sides, but are able to accurately match left sides to left sides and right sides to right sides. Under these conditions, attempts to combine left- and right-sided encounters into a single analysis are problematic because detected individuals can yield a number of possible recorded histories, depending on whether *B* encounters are observable. For example, when presented with the recorded histories *L0* and *0R*, we do not know whether these observations arose from the same animal seen on both occasions, or whether it was indeed two different animals each seen on one occasion. Similarly, when *B* detections are not observed, the recorded histories *L0* and *R0* could arise from a single animal or from two different animals.

Table 33: Latent histories (column 2) and recorded histories (columns 6 and 9) from left- and right-side encounters only (LR) or from left-, right-, and both-side encounters (LRB) with 2 sampling events. Taken from McClintock et al (2013).

i	Latent history (ω_i)	$\Pr(\omega_i \theta)$	LR data type			LRB data type		
			Contributed records (j from i)	j	Recorded history ($\tilde{\omega}_j$)	Contributed records (k from i)	k	Recorded history ($\tilde{\omega}_k$)
1	00	$q_1 q_2$	1	0L	1	0L
2	0L	$q_1 p_2 \delta^L$	1....	2	0R	1.....	2	0R
3	0R	$q_1 p_2 \delta^R$.2....	3	L0	.2.....	3	0B
4	OB	$q_1 p_2 \delta^B$	12....	4	LL	.3.....	4	L0
5	L0	$p_1 \delta^L q_2$.3...	5	R0	.4.....	5	LL
6	LL	$p_1 \delta^L p_2 \delta^L$.4..	6	RR	.5.....	6	LB
7	LR	$p_1 \delta^L p_2 \delta^R$.23...			.24.....	7	R0
8	LB	$p_1 \delta^L p_2 \delta^B$.24..			.6.....	8	RR
9	R0	$p_1 \delta^R q_2$5.			.7.....	9	RB
10	RL	$p_1 \delta^R p_2 \delta^L$	1....5.			1....7.....	10	B0
11	RR	$p_1 \delta^R p_2 \delta^R$6			.8.....	11	BL
12	RB	$p_1 \delta^R p_2 \delta^B$	1....6			.9....	12	BR
13	B0	$p_1 \delta^B q_2$.35.			.10...	13	BB
14	BL	$p_1 \delta^B p_2 \delta^L$.45.			.11...		
15	BR	$p_1 \delta^B p_2 \delta^R$.3..6			.12...		
16	BB	$p_1 \delta^B p_2 \delta^B$.4..6			.13...		

Notes: Contributed records columns show the recorded histories arising from specific latent histories for the LR and LRB data types. For example, latent history 16, $\omega_{16} = BB$, gives rise to recorded histories $j = 4$ and $j = 6$, $\tilde{\omega}_4 = LL$ and $\tilde{\omega}_6 = RR$, for the LR data type. For the LRB data type, $\omega_{16} = BB$ spawns only recorded history $k = 13$, $\tilde{\omega}_{13} = BB$. A zero represents a non-encounter at time t . For illustration, we include the probability of each latent history, $\Pr(\omega | \theta)$, for the closed population abundance model M_t allowing time variation in detection probability p_t ($t = 1, \dots, T$), where $q_t = 1 - p_t$. For the closed population abundance model M_t with $T = 2$, $\theta = (p_1, p_2, \delta^L, \delta^R, N)$, where $\delta^B = 1 - \delta^L - \delta^R$ and N is population abundance. The indices j and k refer to specific recorded histories for the LR and LRB data types, respectively. Other variables are defined in Table 2. This presentation mirrors that of Link et al. (2010).

This is a similar issue as seen when analyzing double tagging experiments with tag loss.

There are two different data types that can be collected:

- only the left or right side (LR data type) or
- the left, right, or both sides (LRB data type). Depending on the design of camera surveys, B encounters may be recorded from a single image (where both sides of an individual are visible), a pair of synchronized images, or multiple images (assuming both sides of an individual are known).

Now for any one particular animal, there are 16 possible (latent) encounter histories that give rise to the observable encounter histories (Table 33).

Column 4 in Table 33 indicates which underlying (latent) encounter histories are mapped to the observable encounter histories. For example, an animal not seen in the first sampling event, and then both sides seen the second sampling event (OB, forth row), gives rise to two observable

(recorded) histories ($0L$ and $0R$) in experiment where both sides can never be matched to a single animal.

The parameters of this model are

- N abundance
- p_t encounter probability at sampling event t
- δ^L probability that an individual which is encountered has its Left side photographed
- δ^R probability that an individual which is encountered has its right side photographed
- δ^B probability that an individual which is encountered has both sides photographed

These parameters can be used to compute the probability of each underlying (latent) encounter history (column 3 in Table 33). These probabilities must sum to 1 over all latent encounter histories. Now many of the latent encounter histories are mapped to multiple recorded histories. Their probabilities will no longer sum to 1 because many latent histories are double counted, and the counts from the reported histories (columns 6 and 9) no longer follow a multinomial distribution.

Because of this potential double counting, a likelihood analysis is very difficult and both Bonner and Holmberg (2013) and McClintock et al (2013) developed a Bayesian approach. Their papers should be consulted for details. McClintock (2015) developed an *R* package (*multimark*) that performs the Bayesian analysis using Monte Carlo Markov Chain (MCMC) methods which will be illustrated below.

By referring to Table 33, the expected counts for each of the capture histories under the LR protocol are:

- $E[n_{L0}] = Np_1(1 - \delta_R)(1 - p_2 + p_2\delta_R)$
- $E[n_{LL}] = Np_1(1 - \delta_R)p_2(1 - \delta_R)$
- $E[n_{0L}] = N(1 - p_1 + p_1\delta_R)p_2(1 - \delta_R)$
- $E[n_{R0}] = Np_1(1 - \delta_L)(1 - p_2 + p_2\delta_L)$
- $E[n_{RR}] = Np_1(1 - \delta_L)p_2(1 - \delta_L)$
- $E[n_{0R}] = N(1 - p_1 + p_1\delta_L)p_2(1 - \delta_L)$

Notice that $\delta_L + \delta_R + \delta_B = 1$.

Two Petersen estimators for abundance can be found by using the animals tagged on the Left or tagged on the Right exclusively, i.e.,

$$\widehat{N}_L = \frac{(n_{L0} + n_{LL})(n_{0L} + n_{LL})}{n_{LL}}$$

$$\widehat{N}_R = \frac{(n_{R0} + n_{RR})(n_{0R} + n_{RR})}{n_{RR}}$$

By substituting in the expected values, we see that these estimators are consistent for N .

The two estimates could be averaged in some fashion, but computing the SE of the combined estimate is complex because of the correlation in the the estimates.

Unfortunately, it is not possible to estimate p_1 , p_2 , δ_L or δ_R individually. But we can estimate, for example

$$p_1(\widehat{1 - \delta_R}) = \frac{n_{LL}}{(n_{LL} + n_{OL})}$$

which is an estimate of “Left side” catchability at time 1.

We can also estimate

$$\frac{(\widehat{1 - \delta_R})}{(1 - \delta_L)} = \frac{(n_{LL} + n_{L0})}{(n_{RR} + n_{R0})}$$

which is ratio of the complement of the conditional probabilities of detection on *Left* or *Right*. But the individual values of δ_L and δ_R cannot be found.

The latter will have implications when using *multimark*. If for example, left or right conditional probabilities are equal (which seems sensible for many species and experimental setups), then any value for $\delta_L = \delta_R$ between 0 and 0.5 gives a ratio of the complements equal to 1. Hence the δ values are non-identifiable individually, and the program will converge to a value dictated by the prior distribution for δ . Similarly, the individual estimates of p_1 and p_2 will depend on the final value for the δ values and may be nonsensical. **But incredibly, estimates of N will be valid!**

This will be illustrated using simulated data.

However, if allow for some animals to have both sides recorded, then not only are the estimates of abundance unbiased, but you also get unbiased estimates of catchability delta.

11.3 Illustration of using *multimark* with simulated data.

11.3.1 LR-type surveys

The *multimark* package has a simulation function to generate data. We will simulate a population of 5000 individuals with different capture probabilities at the two sample times, but equal δ values. Notice that *multimark* uses δ_1 for δ_L , etc. In this simulation, no animals have both sides recorded.

```
logit <- function (p){ log(p/(1-p)) }
expit <- function(theta){ 1/(1+exp(-theta)) }

set.seed(234324)
N= 5000
```

```

p1= .5
p2 = .7

test <- multimark::simdataClosed(
  N = N,
  noccas = 2,
  pbeta = logit(c(p1, p2)),
  tau = 0, # behavioural effects (i.e. c)
  sigma2_zp = 0,
  delta_1 = 0.4,
  delta_2 = 0.4,
  alpha = 0,
  data.type = "never",
  link = "logit"
)

```

The first few encounter histories under the LR protocol are

	[,1]	[,2]
[1,]	0	1
[2,]	2	0
[3,]	1	0
[4,]	0	1
[5,]	1	1
[6,]	2	2
[7,]	1	1
[8,]	0	1
[9,]	1	1
[10,]	0	1

We see that encounter matrix has capture histories with 1 representing a L capture and a 2 representing a R capture.

The reduced set of capture histories is:

```

test.ch <- as.data.frame(test$Enc.Mat)

test.ch.red <- plyr::ddply(test.ch,c("V1","V2"), plyr::summarize,
                           cap_hist=paste0(V1[1],V2[1]),
                           freq=length(V1))
test.ch.red

```

V1	V2	cap_hist	freq
1	0	1	01 1475
2	0	2	02 1453
3	1	0	10 899
4	1	1	11 631
5	2	0	20 900
6	2	2	22 621

Notice that $\delta_B = 1 - \delta_1 - \delta_2 = 0.2$ so for 20% of animals encountered at a sampling event, both the left and right are photographed, but you are unable to link those two photographs together (e.g. taken by different individuals). This implies that the total number of encounters is MORE than expected from the values of p_1 and p_2 by themselves:

Number of pictures of animals at t=1 3051 vs. expected number of 2500

Number of pictures of animals at t=1 4180 vs. expected number of 3500

There evidently some animals detected on both sides, but photos could not be matched. The simulation function also returns the actual encounter history matrix (not shown) that show that some animals are detected on both sides, but could not be matched across the two photographs.

We can get estimates of abundance based on the left or right photos individually in the same way as shown in this document.

For example, based on the left side photographs, the capture history matrix and frequency counts are:

V1	V2	cap_hist	freq
1	0	1	01 1475
3	1	0	10 899
4	1	1	11 631

n1	n2	m2	p.recap	mf
1 1530	2106	631	0.4124183	0.2996201

This gives the estimate of abundance that is consistent, but estimates of catchability that are confounded with the δ values.

```
left.fit <- Petersen::LP_fit(temp, p_model=~..time)
left.est <- Petersen::LP_est(left.fit)
left.est$summary[, c("N_hat", "N_hat_SE", "N_hat_conf_level", "N_hat_LCL", "N_hat_UCL")]
```

```
N_hat N_hat_SE N_hat_conf_level N_hat_LCL N_hat_UCL
1 5106.467 130.4088           0.95 4857.162 5368.568
```

```
left.est$detail$data.expand[1:2,c(..time,"p")]
```

```
..time          p
1             1 0.2996200
1.1           2 0.4124183
```

Similarly, the capture histories and estimated abundance based on the right side photographs is found as:

```
temp <- test.ch.red[grep('2', test.ch.red$cap_hist),]
temp$cap_hist <- gsub('2','1',temp$cap_hist)
temp
```

```
V1 V2 cap_hist freq
2 0 2      01 1453
5 2 0      10 900
6 2 2      11 621
```

```
Petersen::LP_summary_stats(temp)
```

```
n1   n2   m2   p.recap      mf
1 1521 2074 621 0.408284 0.2994214
```

```
right.fit <- Petersen::LP_fit(temp, p_model=~..time)
right.est <- Petersen::LP_est(right.fit)
right.est$summary[, c("N_hat","N_hat_SE","N_hat_conf_level","N_hat_LCL","N_hat_UCL")]
```

```
N_hat N_hat_SE N_hat_conf_level N_hat_LCL N_hat_UCL
1 5079.794 131.2457           0.95 4828.962 5343.656
```

```
right.est$detail$data.expand[1:2,c(..time,"p")]
```

```
..time          p
2             1 0.2994217
2.1           2 0.4082842
```

A naive estimate that combines the two estimates is a simple average with a SE assuming (incorrectly) that these estimates are independent is:

```
# naive simple average
naive.N.avg <- (left.est$summary$N_hat + right.est$summary$N_hat)/2
cat("Naive estimate of abundance based on simple average ", round(naive.N.avg), "\n")

Naive estimate of abundance based on simple average 5093

naive.N.avg.se <- sqrt((left.est$summary$N_hat_SE^2 + right.est$summary$N_hat_SE^2)/4)
cat("Naive SE of average is ", round(naive.N.avg.se), "\n")

Naive SE of average is 93
```

We now use *multimark* to estimate abundance based on the left and right photographs combined. A model that allows for different capture probabilities at each sampling event is fit by setting up a design matrix. The *multimark* package does a Bayesian analysis and so you must specify the number of MCMC iterations to perform etc. The fitting function fits multiple chains in parallel using multiple cores in your machine.

```
# use multimark to estimate abundance

# we set up my own design matrix, because mod.p=~time creates a design matrix
# that has 3 columns rather than only 2
myCovs <- data.frame(mytime=c(0,1))
myCovs

mytime
1      0
2      1

max.iter <- 100000
n.thin   <- 10

multimark.res <- file.path("Images","multimark-sim-LR.Rdata")
if( file.exists(multimark.res)){
  load(multimark.res)
}
if(!file.exists(multimark.res)){
  fit <- multimark::multimarkClosed(
```

```

test$Enc.Mat,
data.type = "never",
covs = myCovs,
mms = NULL,
mod.p = ~mytime,
mod.delta = ~type,
parms = c("pbeta", "delta", "N"),
nchains = 3,
iter = max.iter,
adapt = 10000,
bin = 50,
thin = n.thin,
burnin = 2000)
save(fit, file=multimark.res)
}

```

A summary of the MCMC output is:

```
summary(fit$mcmc)
```

```

Iterations = 2010:1e+05
Thinning interval = 10
Number of chains = 3
Sample size per chain = 9800

```

1. Empirical mean and standard deviation for each variable,
plus standard error of the mean:

	Mean	SD	Naive SE	Time-series SE
pbeta[(Intercept)]	0.4077	5.817e-02	3.393e-04	0.0042681
pbeta[mytime]	1.1306	8.011e-02	4.672e-04	0.0051688
N	5065.2233	1.031e+02	6.013e-01	8.6828559
delta_1	0.5010	6.404e-03	3.735e-05	0.0002343
delta_2	0.4953	6.438e-03	3.755e-05	0.0002414

2. Quantiles for each variable:

	2.5%	25%	50%	75%	97.5%
pbeta[(Intercept)]	0.2948	0.3672	0.4081	0.4485	0.5191
pbeta[mytime]	0.9828	1.0742	1.1278	1.1846	1.2924

```

N          4877.0000 4992.0000 5060.0000 5135.0000 5276.0000
delta_1      0.4883    0.4967    0.5011    0.5054    0.5134
delta_2      0.4824    0.4910    0.4953    0.4997    0.5077

```

```

# get the estimates of p and c
pc <- multimark::getprobsClosed(fit, link = "logit")
summary(pc)

```

```

Iterations = 2010:1e+05
Thinning interval = 10
Number of chains = 3
Sample size per chain = 9800

```

1. Empirical mean and standard deviation for each variable, plus standard error of the mean:

	Mean	SD	Naive SE	Time-series SE
p[1]	0.6005	0.01395	8.134e-05	0.001023
p[2]	0.8225	0.01756	1.024e-04	0.001430
c[2]	0.8225	0.01756	1.024e-04	0.001430

2. Quantiles for each variable:

	2.5%	25%	50%	75%	97.5%
p[1]	0.5732	0.5908	0.6006	0.6103	0.6269
p[2]	0.7881	0.8102	0.8227	0.8352	0.8552
c[2]	0.7881	0.8102	0.8227	0.8352	0.8552

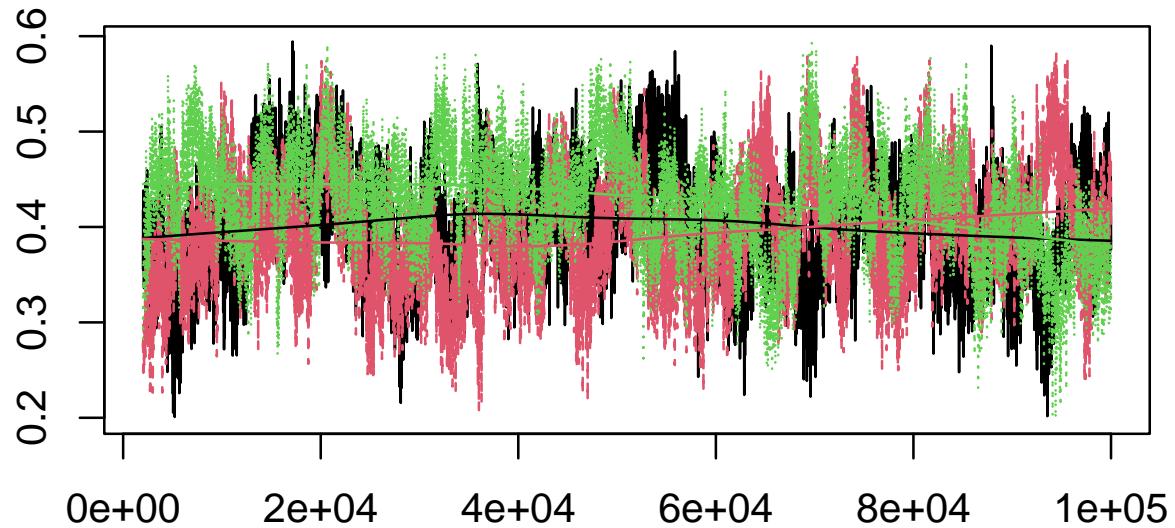
We see that the estimate of abundance appears to be unbiased and has comparable uncertainty to our naive average of estimates of abundance based on the left or right photo individually.

But the estimates of catchability and δ are severely biased. Notice that δ_1 and δ_2 move towards 0.5 based on the prior distribution for these parameters. Consequently because δ_1 is overestimated, $1 - \delta_1$ is under estimated, and so the estimated capture probabilities are overestimated..

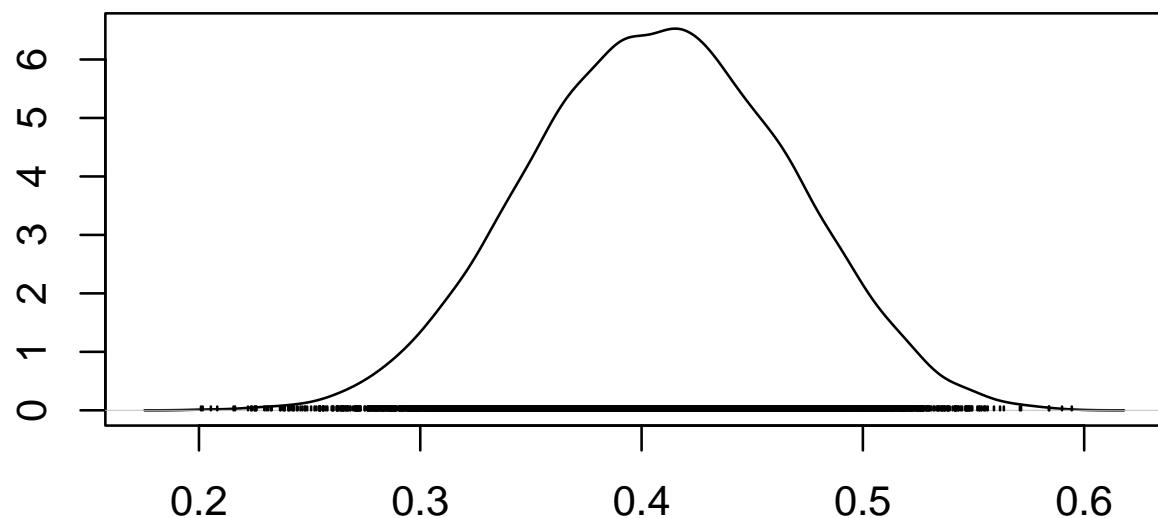
We can look at the diagnostic plots of the MCMC chains:

```
plot(fit$mcmc, auto.layout=FALSE)
```

Trace of $p\beta\alpha[(\text{Intercept})]$

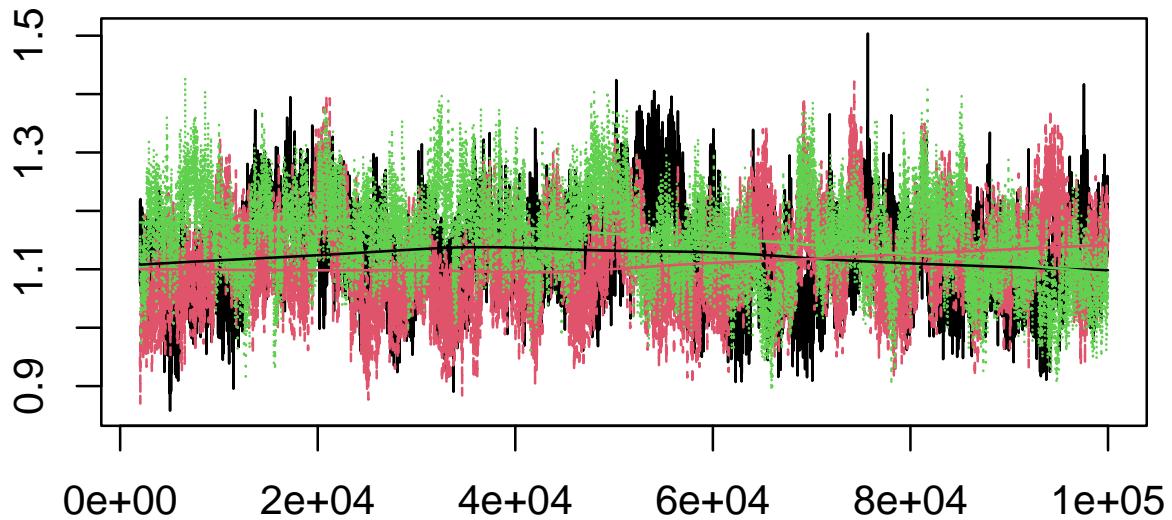


Iterations
Density of $p\beta\alpha[(\text{Intercept})]$

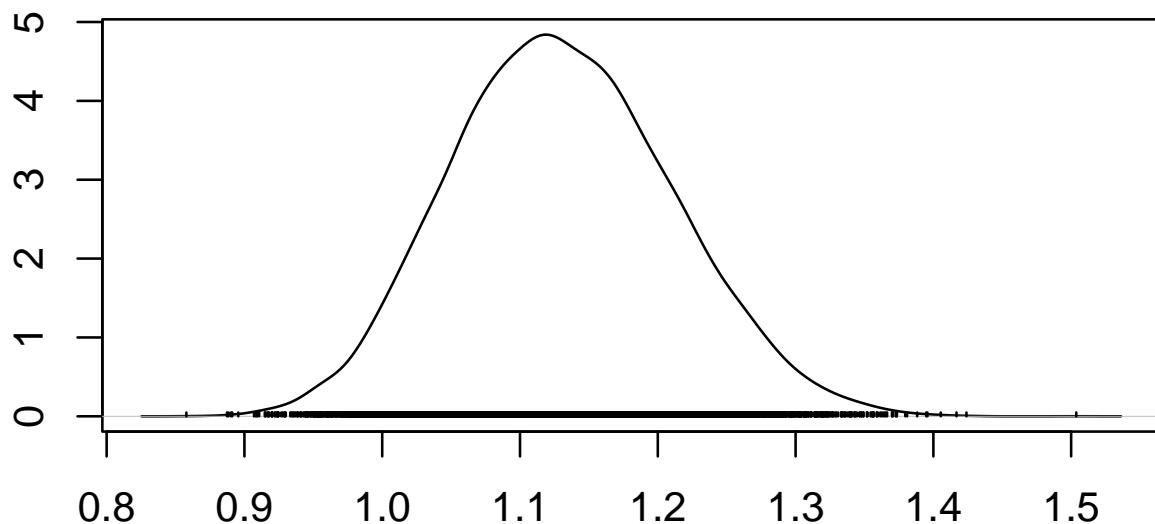


$N = 9800$ Bandwidth = 0.007877

Trace of pbeta[mytime]

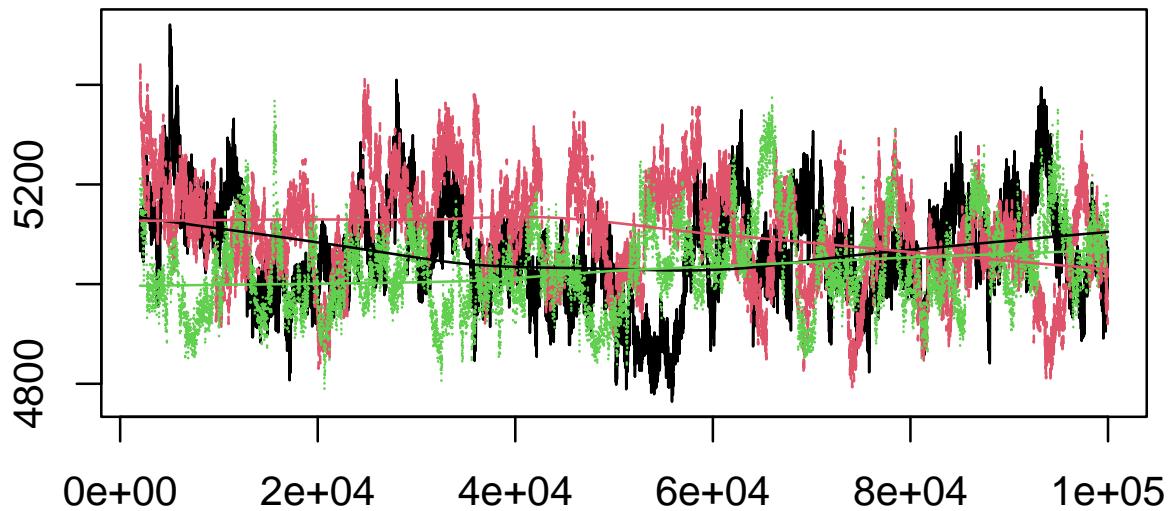


Iterations
Density of pbeta[mytime]

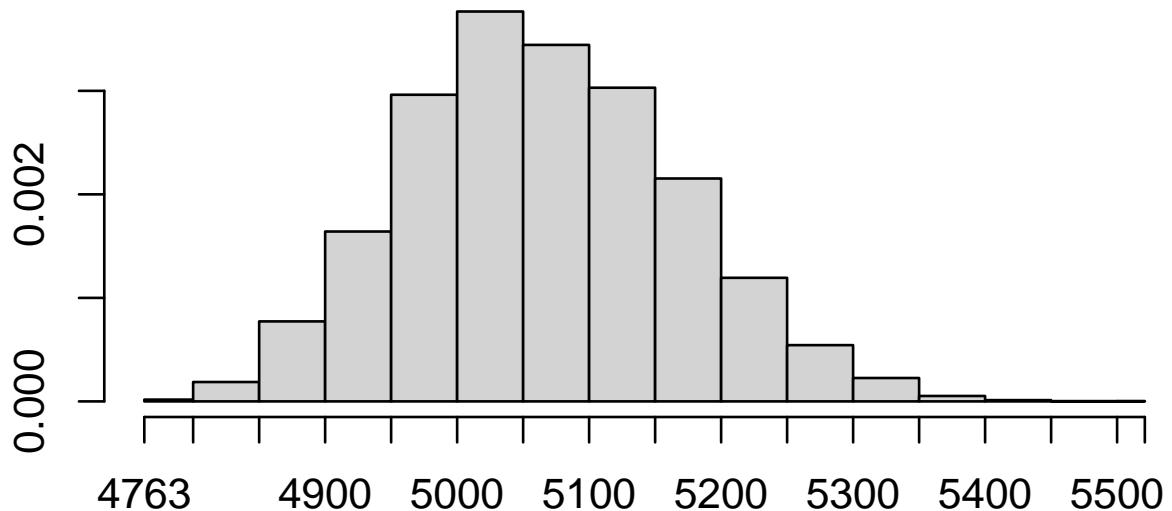


N = 9800 Bandwidth = 0.01085

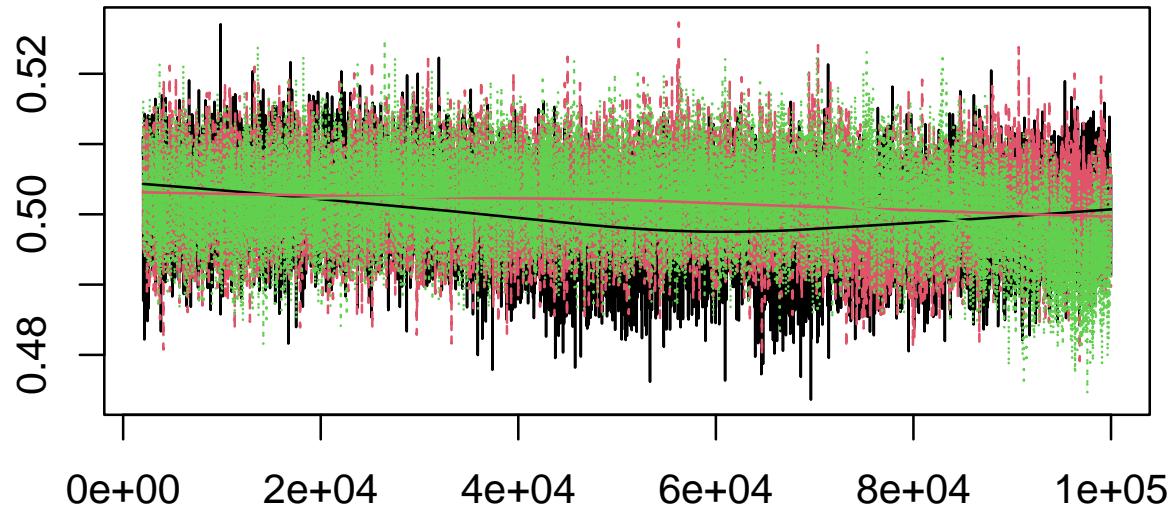
Trace of N



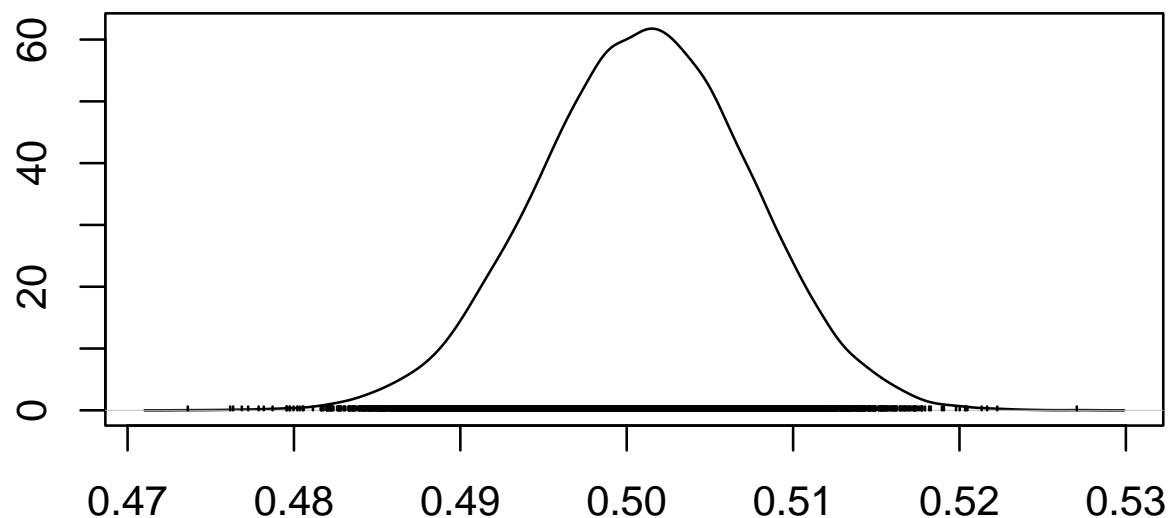
Iterations
Density of N



Trace of δ_1

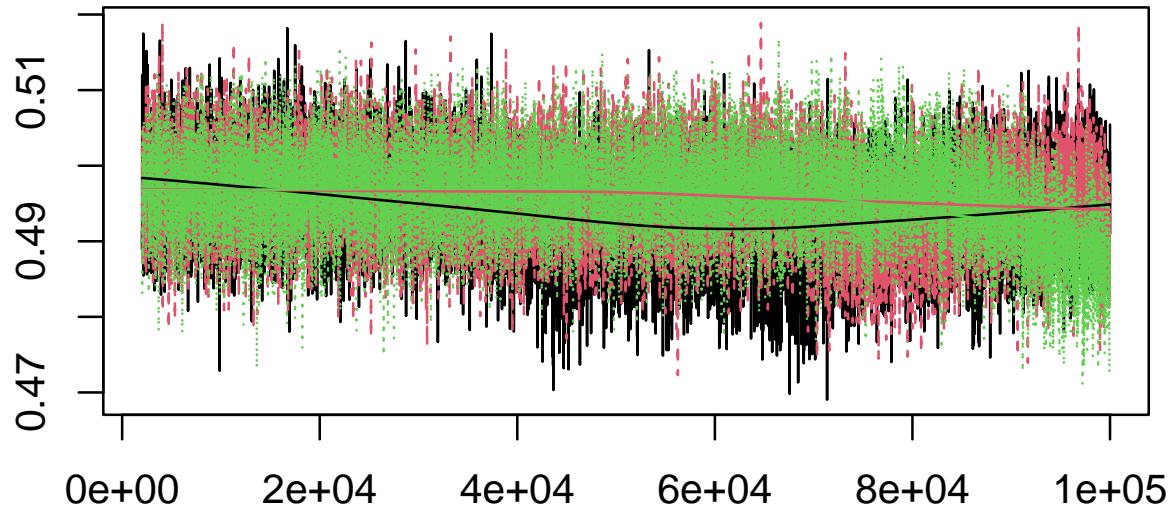


Iterations
Density of δ_1

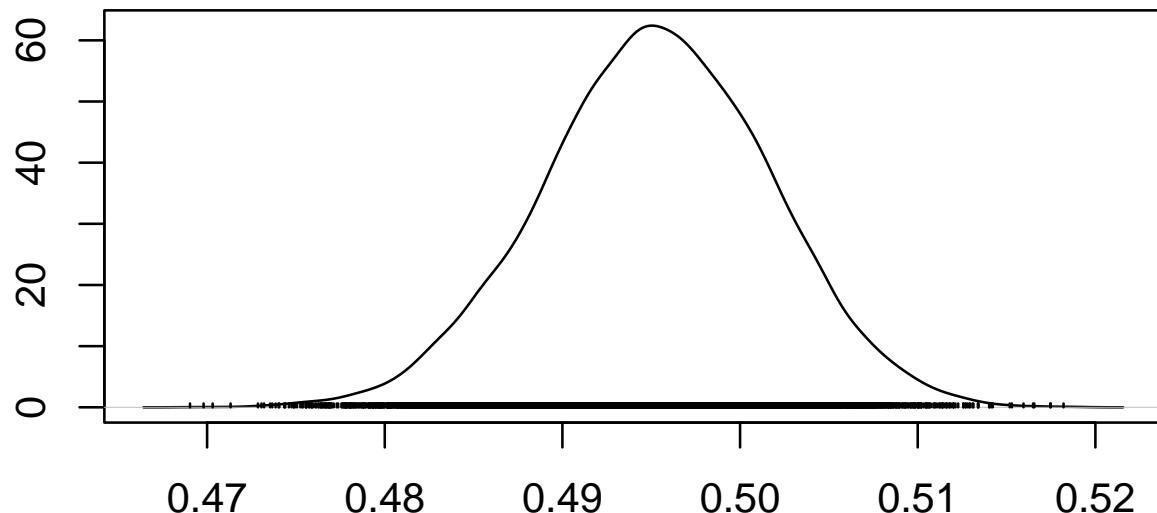


N = 9800 Bandwidth = 0.0008671

Trace of delta_2



Iterations
Density of delta_2



N = 9800 Bandwidth = 0.0008718

Everything appears to be ok, but **only the estimates of abundance are sensible**, and the

estimates of p or δ really have no meaning!

Given these issues of non-identifiability in the case of 2 sampling events, model averaging likely doesn't make much sense.

It is not clear if these issues of identifiability also affect analyses with more than 2 sampling events, but I suspect that they do. Some careful analysis using *multimark* will be required. YMMV.

11.3.2 LRB-type surveys

We now generate simulated data but both sides of some animals can be observed (the B capture type). We will again simulate a population of 5000 individuals with different capture probabilities at the two sample times, but equal δ values. About 0.20 of animals that are encountered have both sides read.

```
logit <- function (p){ log(p/(1-p))} 
expit <- function(theta){ 1/(1+exp(-theta))}

set.seed(234324)
N= 5000
p1= .5
p2 = .7

test <- multimark::simdataClosed(
  N = N,
  noccas = 2,
  pbeta = logit(c(p1, p2)),
  tau = 0, # behavioural effects (i.e. c)
  sigma2_zp = 0,
  delta_1 = 0.4,
  delta_2 = 0.4,
  alpha = 0,
  data.type = "always", # some animals have both sides read.
  link = "logit"
)
```

The first few encounter histories under the LRB protocol are

```
[,1] [,2]
[1,]    0    1
[2,]    2    0
```

```
[3,] 1 0
[4,] 0 1
[5,] 4 1
[6,] 2 2
[7,] 4 1
[8,] 0 4
[9,] 4 1
[10,] 2 4
```

We see that encounter matrix has capture histories with *1* representing a *L* capture and a *2* representing a *R* capture. A capture history with a *4* represents an animal that has both sides read.

The reduced set of capture histories is:

```
test.ch <- as.data.frame(test$Enc.Mat)

test.ch.red <- plyr::ddply(test.ch,c("V1","V2"), plyr::summarize,
                           cap_hist=paste0(V1[1],V2[1]),
                           freq=length(V1))
test.ch.red
```

	V1	V2	cap_hist	freq
1	0	1	01	993
2	0	2	02	972
3	0	4	04	351
4	1	0	10	582
5	1	1	11	275
6	1	4	14	130
7	2	0	20	596
8	2	2	22	251
9	2	4	24	131
10	4	0	40	154
11	4	1	41	150
12	4	2	42	163
13	4	4	44	76

Notice that $\delta_B = 1 - \delta_1 - \delta_2 = 0.2$ so for 20% of animals encountered at a sampling event, both the left and right are photographed and the two sides can be linked. Some animals with both sides linked are also seen only on the left or right side.

Unlike before, this implies that the total number of encounters is roughly equal to that expected from the values of p_1 and p_2 by themselves because any animal that has both sides captures has them linked.

```
Number of pictures of animals at t=1 2508 vs. expected number of 2500
```

```
Number of pictures of animals at t=1 3492 vs. expected number of 3500
```

Now any animal that has both sides photographed has them linked.

We can get estimates of abundance based on the left or right photos individually in the same way as shown in this document. We note that type 4 encounters are used in both the left and right sides estimates.

For example, based on the left side photographs, the capture history matrix and frequency counts are:

V1	V2	cap_hist	freq
1	0	01	993
3	0	01	351
4	1	10	582
5	1	11	275
6	1	11	130
9	2	01	131
10	4	0	154
11	4	1	150
12	4	2	163
13	4	11	76

n1	n2	m2	p.recap	mf
1	1530	2106	631	0.4124183 0.2996201

This gives the estimate of abundance that is consistent, but estimates of catchability that are confounded with the δ values.

```
left.fit <- Petersen::LP_fit(temp, p_model=~..time)
left.est <- Petersen::LP_est(left.fit)
left.est$summary[, c("N_hat", "N_hat_SE", "N_hat_conf_level", "N_hat_LCL", "N_hat_UCL")]
```

N_hat	N_hat_SE	N_hat_conf_level	N_hat_LCL	N_hat_UCL
1	5106.467	130.4088	0.95	4857.162 5368.568

```
left.est$detail$data.expand[1:2,c(..time","p")]
```

```

..time          p
1           1 0.2996200
1.1         2 0.4124183

```

Similarly, the capture histories and estimated abundance based on the right side photographs is found as:

```

temp <- test.ch.red
temp$cap_hist <- gsub('4','2',temp$cap_hist)
temp$cap_hist <- gsub('1','0',temp$cap_hist)
temp$cap_hist <- gsub('2','1',temp$cap_hist)
temp <- temp[grep('1', temp$cap_hist),]
temp

```

V1	V2	cap_hist	freq
2	0	2	01 972
3	0	4	01 351
6	1	4	01 130
7	2	0	10 596
8	2	2	11 251
9	2	4	11 131
10	4	0	10 154
11	4	1	10 150
12	4	2	11 163
13	4	4	11 76

```
Petersen::LP_summary_stats(temp)
```

```

n1   n2   m2   p.recap      mf
1 1521 2074 621 0.408284 0.2994214

```

```

right.fit <- Petersen::LP_fit(temp, p_model=~..time)
right.est <- Petersen::LP_est(right.fit)
right.est$summary[, c("N_hat", "N_hat_SE", "N_hat_conf_level", "N_hat_LCL", "N_hat_UCL")]

```

```

N_hat N_hat_SE N_hat_conf_level N_hat_LCL N_hat_UCL
1 5079.794 131.2457          0.95 4828.962 5343.656

```

```
right.est$detail$data.expand[1:2,c(..time","p")]
```

```
..time      p
2          1 0.2994217
2.1        2 0.4082842
```

A naive estimate that combines the two estimates is a simple average with a SE assuming (incorrectly) that these estimates are independent (which is incorrect because some animals are involved in both estimates) is:

```
# naive simple average
naive.N.avg <- (left.est$summary$N_hat + right.est$summary$N_hat)/2
cat("Naive estimate of abundance based on simple average ", round(naive.N.avg), "\n")
```

```
Naive estimate of abundance based on simple average 5093
```

```
naive.N.avg.se <- sqrt((left.est$summary$N_hat_SE^2 + right.est$summary$N_hat_SE^2)/4)
cat("Naive SE of average is ", round(naive.N.avg.se), "\n")
```

```
Naive SE of average is 93
```

We now use *multimark* to estimate abundance based on the left and right photographs combined. A model that allows for different capture probabilities at each sampling event is fit by setting up a design matrix. The *multimark* package does a Bayesian analysis and so you must specify the number of MCMC iterations to perform etc. The fitting function fits multiple chains in parallel using multiple cores in your machine.

```
# use multimark to estimate abundance

# we set up my own design matrix, because mod.p=~time creates a design matrix
# that has 3 columns rather than only 2
```

```
myCovs <- data.frame(mytime=c(0,1))
myCovs
```

```
mytime
1      0
2      1
```

```

max.ITER <- 100000
n.thin   <- 10

multimark.res <- file.path("Images", "multimark-sim-LRB.Rdata")
if( file.exists(multimark.res)){
  load(multimark.res)
}
if(!file.exists(multimark.res)){
  fit <- multimark::multimarkClosed(
    test$Enc.Mat,
    data.type = "always",
    covs = myCovs,
    mms = NULL,
    mod.p = ~mytime,
    mod.delta = ~type,
    parms = c("pbeta", "delta", "N"),
    nchains = 3,
    iter = max.ITER,
    adapt = 10000,
    bin = 50,
    thin = n.thin,
    burnin = 2000)
  save(fit, file=multimark.res)
}

```

A summary of the MCMC output is:

```
summary(fit$mcmc)
```

```

Iterations = 2010:1e+05
Thinning interval = 10
Number of chains = 3
Sample size per chain = 9800

```

1. Empirical mean and standard deviation for each variable,
plus standard error of the mean:

	Mean	SD	Naive SE	Time-series SE
pbeta[(Intercept)]	-0.01547	0.043726	2.550e-04	7.549e-04
pbeta[mytime]	0.81855	0.046360	2.704e-04	4.898e-04

N	5057.90820	85.525972	4.988e-01	1.798e+00
delta_1	0.40083	0.006339	3.697e-05	3.697e-05
delta_2	0.39397	0.006316	3.683e-05	3.665e-05

2. Quantiles for each variable:

	2.5%	25%	50%	75%	97.5%
pbeta[(Intercept)]	-0.1012	-0.04517	-0.01525	1.393e-02	7.082e-02
pbeta[mytime]	0.7287	0.78716	0.81828	8.493e-01	9.105e-01
N	4895.0000	4999.00000	5057.00000	5.114e+03	5.228e+03
delta_1	0.3884	0.39657	0.40083	4.051e-01	4.133e-01
delta_2	0.3817	0.38971	0.39392	3.982e-01	4.063e-01

```
# get the estimates of p and c
pc <- multimark::getprobsClosed(fit, link = "logit")
summary(pc)
```

Iterations = 2010:1e+05
 Thinning interval = 10
 Number of chains = 3
 Sample size per chain = 9800

1. Empirical mean and standard deviation for each variable,
 plus standard error of the mean:

	Mean	SD	Naive SE	Time-series SE
p[1]	0.4961	0.01093	6.372e-05	0.0001886
p[2]	0.6905	0.01335	7.787e-05	0.0002506
c[2]	0.6905	0.01335	7.787e-05	0.0002506

2. Quantiles for each variable:

	2.5%	25%	50%	75%	97.5%
p[1]	0.4747	0.4887	0.4962	0.5035	0.5177
p[2]	0.6645	0.6815	0.6904	0.6995	0.7164
c[2]	0.6645	0.6815	0.6904	0.6995	0.7164

We see that the estimate of abundance appears to be unbiased and has comparable uncertainty to our naive average of estimates of abundance based on the left or right photo individually.

But now with the *LRB* study design, estimates of catchability and the delta's are not unbiased.

The MCMC diagnostic plots look fine (but are not shown here)
YMMV.

11.4 Bobcat example

The *multimark* package also ships with a sample dataset on a study of bobcats. Images of the left and right side were captured using camera traps in 8 sampling locations. It is not possible to match left and right photographs of the bobcats.

The first few records, with one line per photograph are:

```
# Get the bobcat data
data("bobcat")
head(bobcat)
```

	occ1	occ2	occ3	occ4	occ5	occ6	occ7	occ8
ID2	0	0	0	0	0	1	1	0
ID3	0	0	1	0	1	0	0	0
ID4	0	0	0	0	1	0	0	0
ID6	1	0	0	0	0	0	0	0
ID7	0	0	1	0	0	0	0	1
ID8	0	1	0	0	0	0	0	0

The data are very sparse.

We collapse the first 4 sampling occasions into the first “event”, and the last 4 sampling occasions into the second “event”. In most cases, the photo was only seen once in each of the 4 sets of occasions.

The reduced capture events are shown at the right of each line below:

```
# classify into 2 events; first 4 sampling events;
# last 4 sampling events; if seen in any of the multiple occasions in an event

bobcat <- cbind(bobcat, V1=apply(bobcat[,c("occ1","occ2","occ3","occ4")],1,max))
bobcat <- cbind(bobcat, V2=apply(bobcat[,c("occ5","occ6","occ7","occ8")],1,max))
head(bobcat)
```

	occ1	occ2	occ3	occ4	occ5	occ6	occ7	occ8	V1	V2
ID2	0	0	0	0	0	1	1	0	0	1
ID3	0	0	1	0	1	0	0	0	1	1
ID4	0	0	0	0	1	0	0	0	0	1

ID6	1	0	0	0	0	0	0	1	0
ID7	0	0	1	0	0	0	1	1	1
ID8	0	1	0	0	0	0	0	1	0

This gives us a set of capture histories:

	V1	V2	cap_hist	freq
1	0	1	01	11
2	0	2	02	13
3	1	0	10	6
4	1	1	11	6
5	2	0	20	6
6	2	2	22	4

The estimated abundances from the left and right sided photographs are:

Left sided abundance estimates

	N_hat	N_hat_SE	N_hat_conf_level	N_hat_LCL	N_hat_UCL
1	33.99992	7.895085		0.95	21.56856 53.59626

Right sided abundance estimates

	N_hat	N_hat_SE	N_hat_conf_level	N_hat_LCL	N_hat_UCL
1	42.5	14.39401		0.95	21.88275 82.54221

Naive estimate of abundance based on simple average 38.2

Naive SE of average is 8.2

We now use *multimark* on the reduced capture histories:

```
# we set up my own design matrix, because mod.p=~time creates a
# design matrix that has 3 columns rather than only 2
myCovs <- data.frame(mytime=c(0,1))
#myCovs

max.iter <- 100000
n.thin   <- 10
```

```

multimark.bobcat.res <- file.path("Images","multimark-bobcat.Rdata")
if( file.exists(multimark.bobcat.res)){
  load(multimark.bobcat.res)
}
if(!file.exists(multimark.bobcat.res)){
  bobcat.fit <- multimark::multimarkClosed(
    as.matrix(bobcat[,c("V1","V2")]),
    data.type = "never",
    covs = myCovs,
    mms = NULL,
    mod.p = ~mytime,
    mod.delta = ~type,
    parms = c("pbeta", "delta", "N"),
    nchains = 3,
    iter = max.iter,
    adapt = 10000,
    bin = 50,
    thin = n.thin,
    burnin = 2000)
  save(bobcat.fit, file=multimark.bobcat.res)
}

```

This gives the following results

```
summary(bobcat.fit$mcmc)
```

```

Iterations = 2010:1e+05
Thinning interval = 10
Number of chains = 3
Sample size per chain = 9800

```

1. Empirical mean and standard deviation for each variable,
plus standard error of the mean:

	Mean	SD	Naive SE	Time-series SE
pbeta[(Intercept)]	-0.2815	0.5679	0.0033118	0.019788
pbeta[mytime]	0.8428	0.6247	0.0036431	0.016251
N	37.5916	8.8429	0.0515727	0.092755
delta_1	0.2654	0.1382	0.0008059	0.008289
delta_2	0.2151	0.1428	0.0008328	0.008920

2. Quantiles for each variable:

	2.5%	25%	50%	75%	97.5%
pbeta[(Intercept)]	-1.34656	-0.6682	-0.2990	0.08557	0.8857
pbeta[mytime]	-0.21701	0.4203	0.7883	1.19137	2.2800
N	26.00000	32.0000	36.0000	41.00000	60.0000
delta_1	0.05044	0.1512	0.2506	0.37023	0.5403
delta_2	0.01011	0.0916	0.1958	0.32458	0.5009

```
# get the estimates of p and c
pc <- multimark::getprobsClosed(bobcat.fit, link = "logit")
summary(pc)
```

Iterations = 2010:1e+05
Thinning interval = 10
Number of chains = 3
Sample size per chain = 9800

1. Empirical mean and standard deviation for each variable,
plus standard error of the mean:

	Mean	SD	Naive SE	Time-series SE
p[1]	0.4344	0.1305	0.0007611	0.004585
p[2]	0.6137	0.1690	0.0009855	0.007349
c[2]	0.6137	0.1690	0.0009855	0.007349

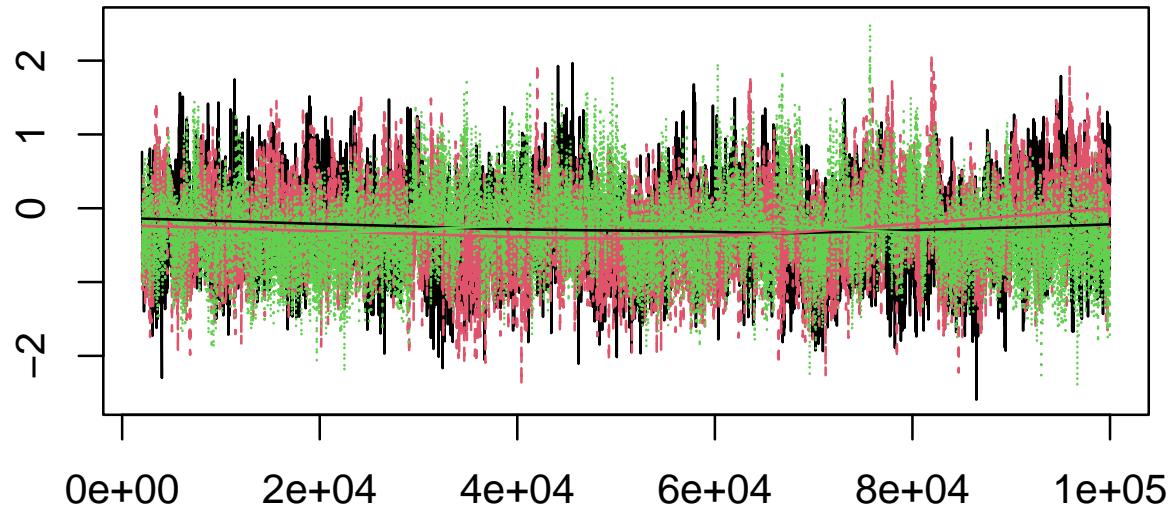
2. Quantiles for each variable:

	2.5%	25%	50%	75%	97.5%
p[1]	0.2064	0.3389	0.4258	0.5214	0.7080
p[2]	0.2975	0.4894	0.6100	0.7358	0.9357
c[2]	0.2975	0.4894	0.6100	0.7358	0.9357

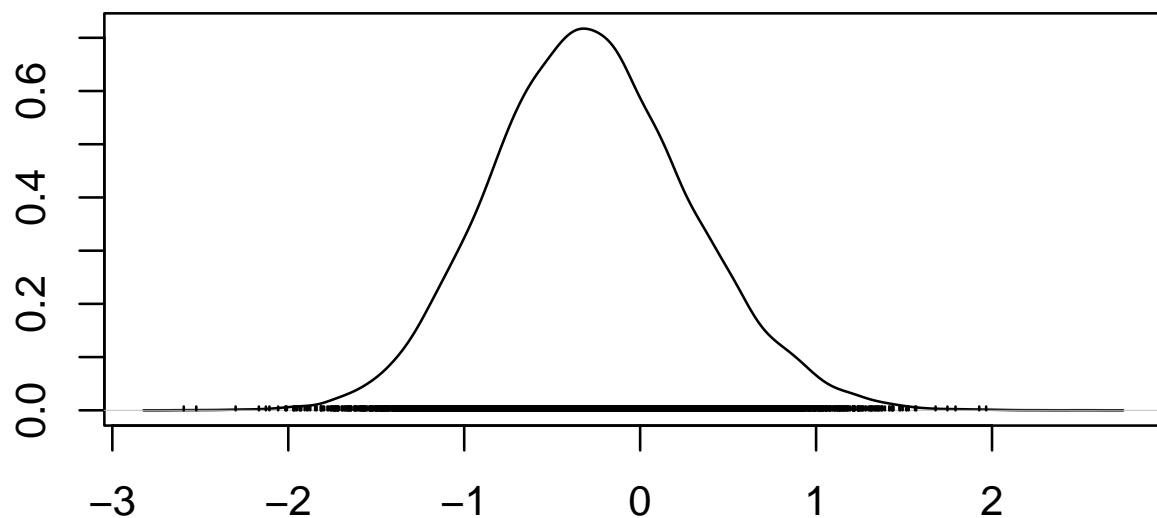
The estimated abundance is comparable to our naive estimate. **It is unclear how the estimates of catchability and for δ perform** (see previous section).

The diagnostic plots from the MCMC iterations are:

Trace of $p\beta\alpha$ [(Intercept)]

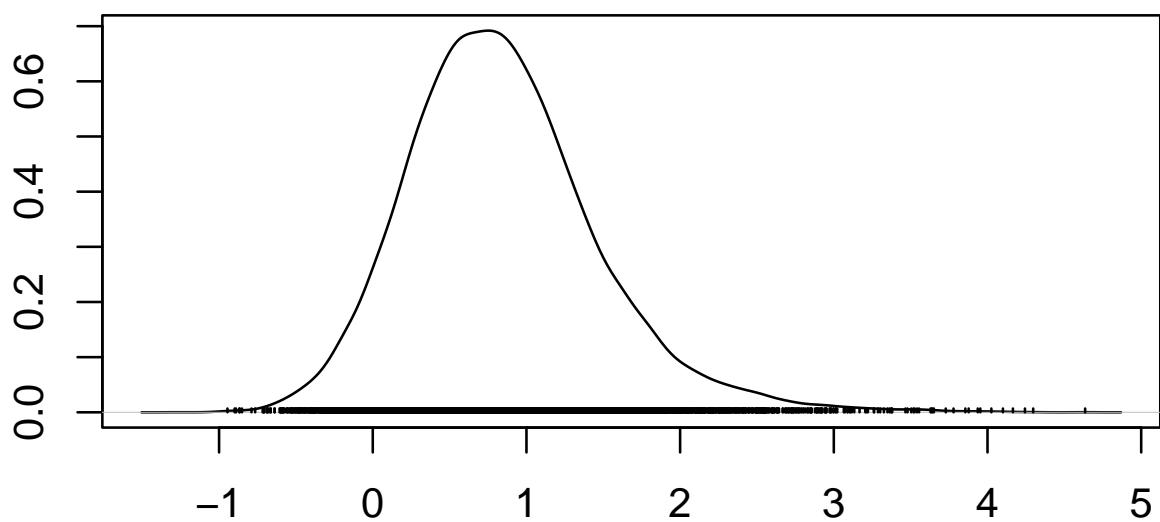
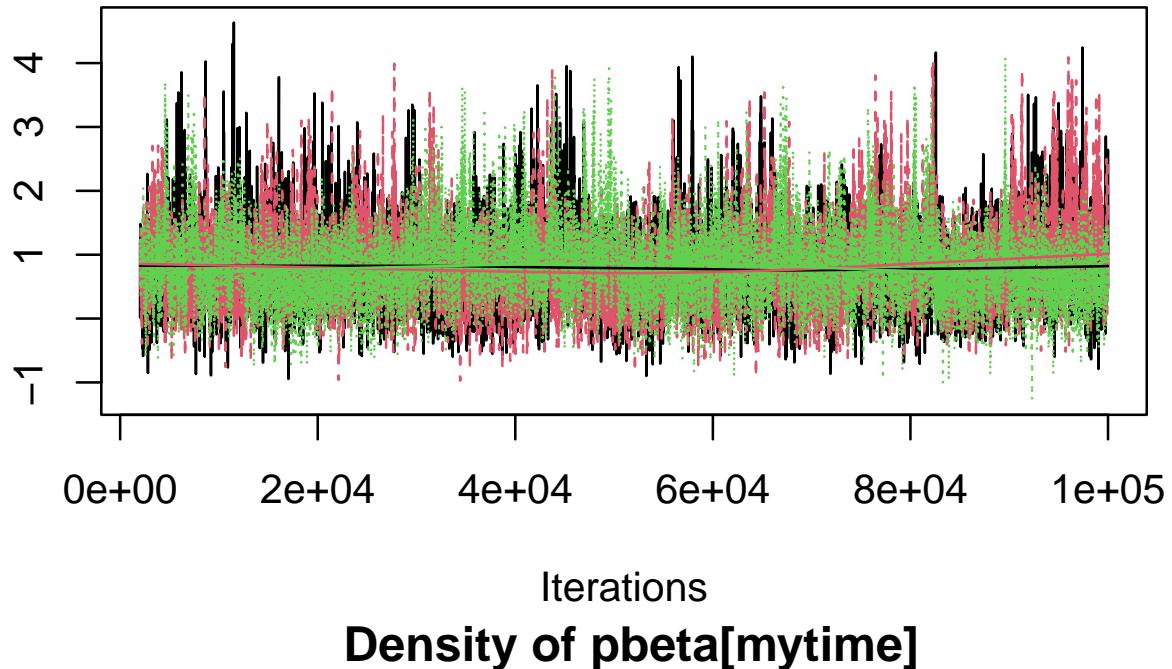


Iterations
Density of $p\beta\alpha$ [(Intercept)]



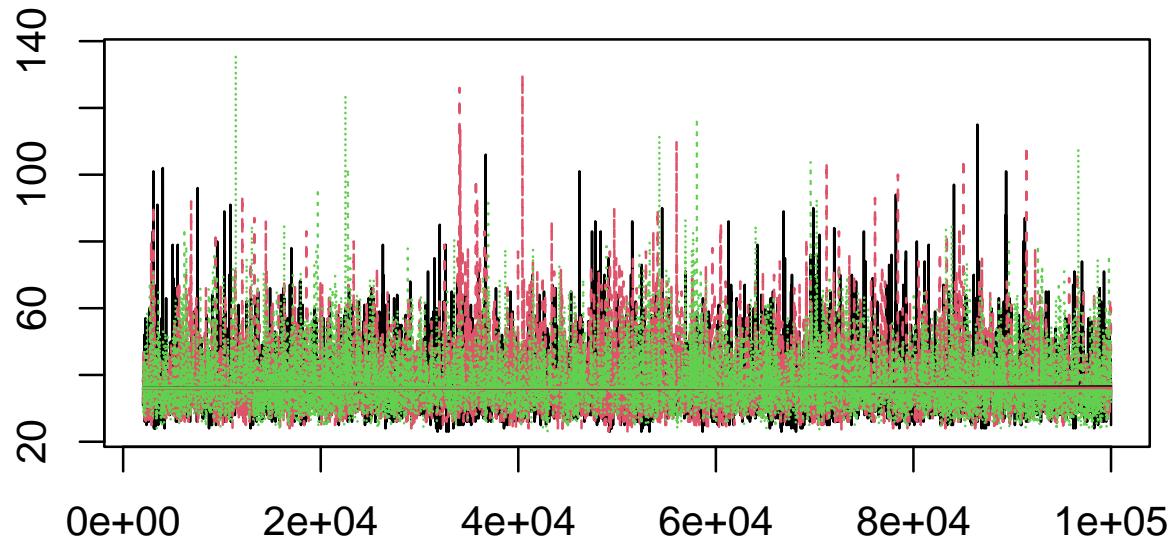
N = 9800 Bandwidth = 0.07617

Trace of pbeta[mytime]

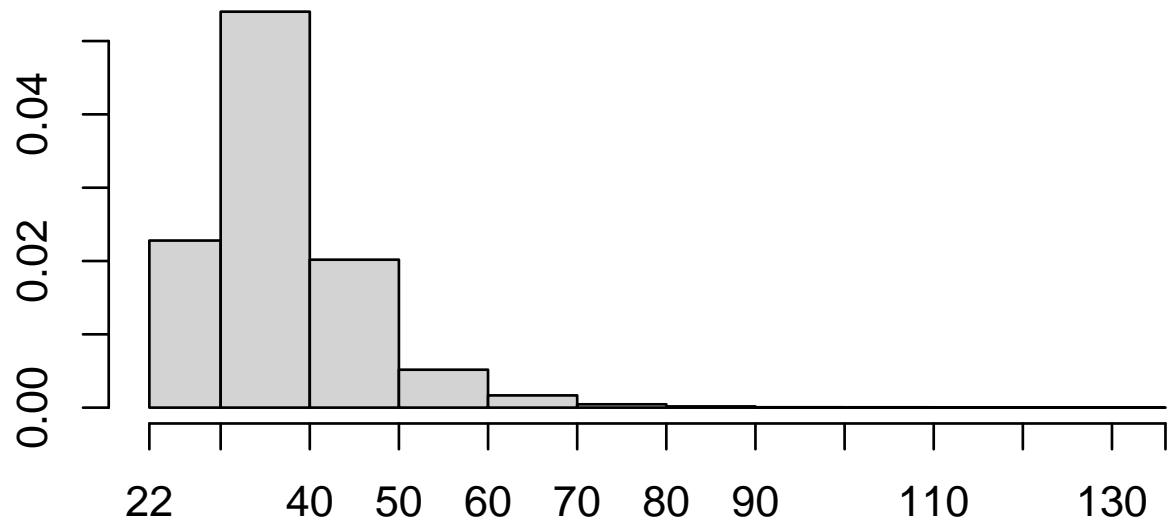


N = 9800 Bandwidth = 0.07792

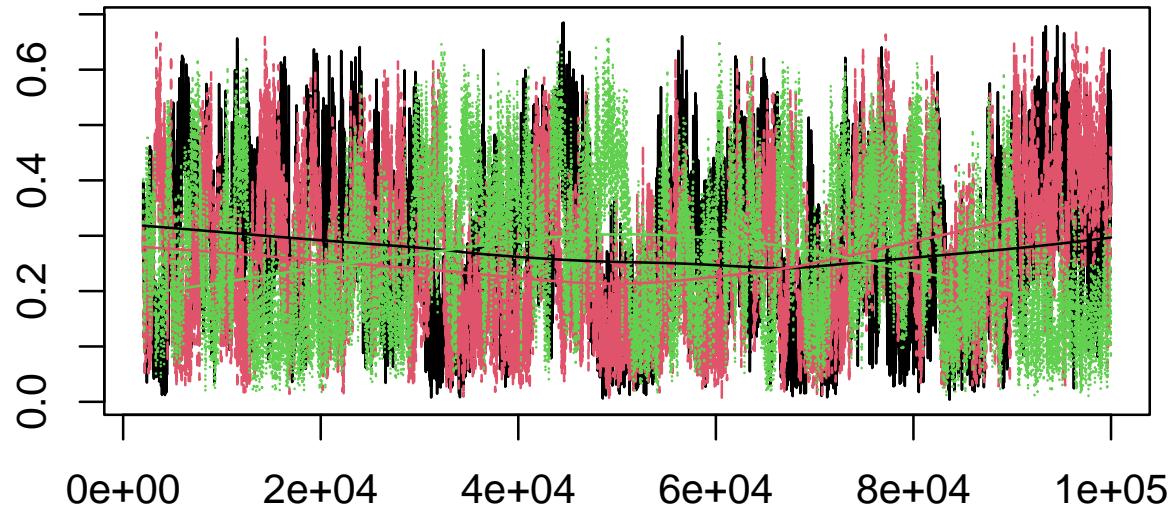
Trace of N



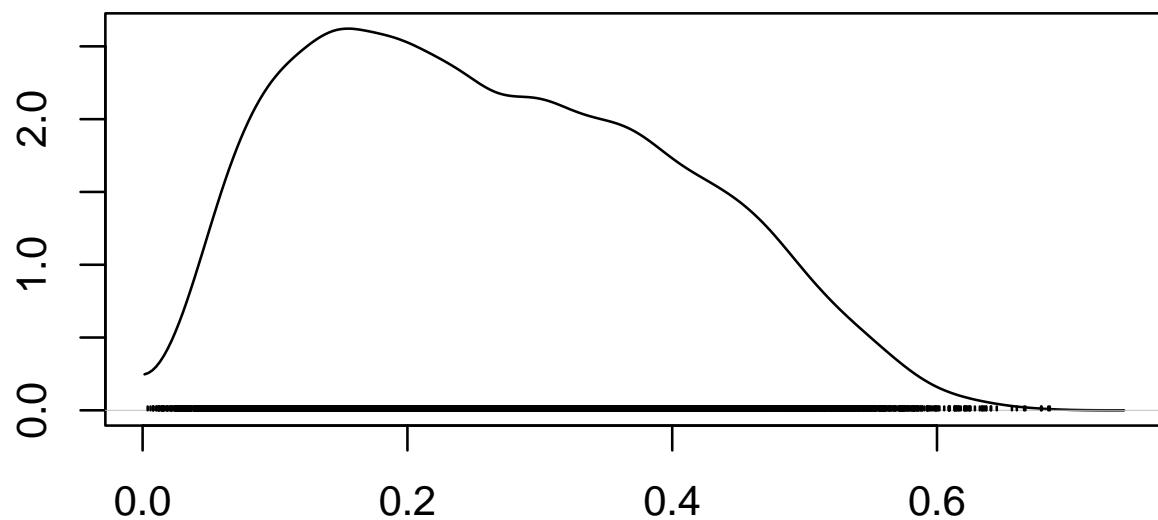
Iterations Density of N



Trace of δ_1

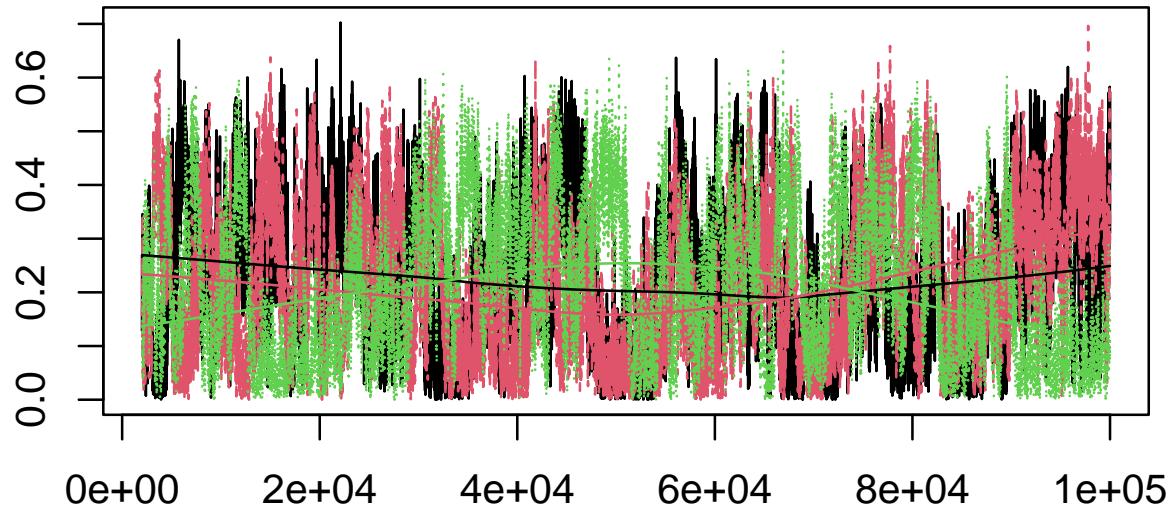


Iterations
Density of δ_1

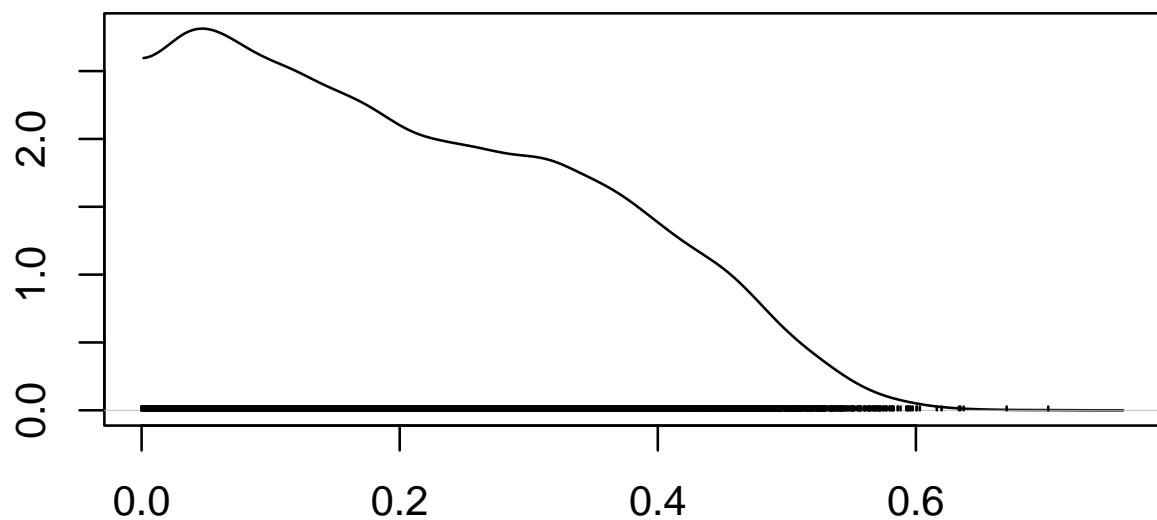


N = 9800 Bandwidth = 0.01871

Trace of delta_2



Iterations
Density of delta_2



N = 9800 Bandwidth = 0.01933

12 Multiple Petersen estimates

12.1 Introduction

In some cases, multiple Petersen experiments occur on study population over a short time period (when the population is assumed closed). For example, 200 fish could be initially marked as they return to their spawning grounds. At a weir, a sample of 1000 fish are taken and the number of tagged fish noted. No additional tags are applied; no tags are removed; and fish are returned to the stream. Then on the spawning ground an additional 1200 fish are sampled of which some are marked.

On the surface, this appears that a closed population model with three sampling occasions could be fit. However, this is not possible because unmarked fish captured at the weir are not tagged and returned to the population. Consequently, no capture histories can be constructed for fish first captured at the weir and released. If individually identifiable tags are used, then a capture-history could be constructed for tagged fish; but if batch marks are used, then this is also not possible because it is impossible to determine if an untagged fish is captured at the weir and then again on the spawning grounds. Finally, it sometimes occurs that additional tags are applied to some (but not all untagged fish) captured at the weir and most closed population models cannot deal with the addition of more tags partway through the study.

These types of studies are known as *mark-resight* models (McClintock et al., 2009; McClintock and White, 2012). There are many previous estimators for this situation - refer to McClintock and White (2012) for details - but modern approaches use the likelihood based estimators where the probability of recapture varies among the sampling occasions, the population is closed, and the number of marks available is known for each sampling occasion. McClintock et al (2012) indicated that the logit-normal estimator should only be used with sampling without replacement so that an animal can only be recaptured a maximum of one time, and the Poisson-log-normal estimator should be used if sampling occurs with replacement, i.e., captured-fish are returned to the study population. However, the latter model will require individually-identifiable tags so that the number of times a tagged-fish is recaptured is known. If batch marks are used and the sampling fractions are small, the logit-normal estimator will provide a close approximation.

McClintock and White (2012) provide a summary of the conditions under which these two (and a third) estimators can be used in Table 34.

Table 34: Comparison of assumptions for the mark-resight estimators in Program MARK

Estimator	Geographic closure	Sampling with replacement	Known number of marks	Identifiable marks
LNE	Required	Not allowed	Required	Not required
PNE	Required	Allowed	Not required	Required
IELNE	Not required	No allowed	Required	Not required

Estimator	Geographic closure	Sampling with replacement	Known number of marks	Identifiable marks
-----------	--------------------	---------------------------	-----------------------	--------------------

The IELNE allows for non-closure of the population (e.g., new animals entering the population), but is not discussed here.

Note that close approximation to the LNE estimator can be obtained by pooling all of the subsequent samples (assuming that the number of marks has not changed), and treating the weir and spawning group samples like a single “larger” recapture event. Duplicate recaptures of marked or unmarked fish are treated as two separate recaptures.

The easiest way to use these models is via *RMark/MARK* as shown below for the LNE model. The PNE model is specified similarly, but requires individually-marked fish.

12.2 Batch marks (logit-normal model)

This model is suitable for batch marks where a capture-history cannot be determined for individual animals. It assumes that sampling is without replacement, but as long as the number of animals sampled is small relative to the population size, the estimator should perform well.

We consider a study where salmon enter a stream and are marked. Two sample are taken upstream, at a weir and on the spawning grounds as summarized in Table 35.

Table 35: Summary of data from a study with two sampling occasions and batch marks

Event	Marks available	Recapture size	Marks seen	Petersen Est
1	n_1 200	n_2 1027	m_2 18	10,874
2	200	1305	26	9,721

We first create the capture-history file with *pseudo*-histories for the tagged fish ignoring the first capture event where 200 marks are applied.

```
data_salmon <- data.frame(
  ch= c("10", "01", "00"),
  freq=c( 18,    26,  200-18-26)
)
data_salmon
```

```

ch freq
1 10   18
2 01   26
3 00   156

```

Notice the first two rows report the number of marks seen at each sampling event. The last history (“00”) is technically unobservable and is a “dummy” history to represent the number of marks originally applied and never seen. Again, we are assuming that the sample sizes are small relative to the population sizes that sampling without replacement is a suitable approximation. The rule needed to create the histories is that the sum of the 1’s in a column x the count must equal the number of marks seen. There are several ways in which the histories can be constructed. The above histories are more easily interpreted if a “1” is temporarily prepended to each history for the marking event.

A . (period) can be used if the number of marks varies among the sampling events (e.g. marks added between event or marks lost between events). The following capture_history file has the same number of recaptures, but 10 **marked fish** were lost after sampling at the weir due to the proximity of a BBQ. Consequently, the number of marks available is 200 and 190 for the two future sampling events.

```

data_salmon2 <- data.frame(
  ch= c("10", "01", "0.", "00"),
  freq=c( 18,    26,     10,  200-18-26-10)
)
data_salmon2

```

```

ch freq
1 10   18
2 01   26
3 0.   10
4 00   146

```

Now there are $200 - 18 - 26 - 10 = 146$ marked fish not seen at both sampling occasions, with an additional 10 fish not seen at the first subsequent sampling occasion, but not available for the second subsequent sampling event. Again, prepend a “1” in front of each history to interpret the histories more readily. Histories can be constructed for adding new marks as well.

We launch *RMark* and see which parameters must be specified for this model:

```

library(RMark)

setup.parameters("LogitNormalMR", check=TRUE)

```

```
[1] "p"      "sigma"  "N"
```

The parameter

- p represents the mean (logit) capture probability;
- σ represent variation in the (logit) capture probability over individuals. Because we don't have individually marked fish, we must set this parameter to 0 for this example.
- N represents the population abundance.

We need to use the `process.data()` function to specify the umber of unmarked fish seen in the experiment:

```
salmon.proc <- process.data(data_salmon, model="LogitNormalMR",
                           counts=list("Unmarked seen"=c(2288)),
                           time.intervals=c(0))
salmon.ddl   <- make.design.data(salmon.proc)
```

The time intervals of 0 indicate a single session (i.e. year) of data where the population is assume closed with two secondary sessions in this single primary session. The `make.design.data()` function creates the analysis structure.

Specify a model in the usual way using *RMark*. We will start with a $p(t)$, $\sigma=0$ model.

Due to a ‘feature’ in *MARK* when called by *RMark*, poor initial values are chosen and the model converges to nonsense. We need to create *sensible* initial values on the β scale. For the capture probabilities (the logit scale), we can use the value of 0 (once for each occasion), for the sigma parameter, we are initializing and fixing at 0 and for the abundance (N), we use $\log(\text{approximate abundance})$. Here the approximate abundance is about 10,000 so we use $\log(10,000)$.

```
salmon.mt <- mark( salmon.proc,
                     model="LogitNormalMR",
                     model.name="(p(t) sigma=0)",
                     model.parameters=
                     list(p     =list(formula=~time),
                          sigma=list(formula=~1, fixed=0),
                          N      =list(formula=~1)),
                     initial=c(0,0,0,log(10000))
                   )
```

```
Output summary for LogitNormalMR model
Name : (p(t) sigma=0)
```

```
Npar : 3  
-2lnL: 285.0223  
AICc : 291.0828
```

```
Beta
```

	estimate	se	lcl	ucl
p:(Intercept)	-2.3127836	0.2472701	-2.7974330	-1.828134
p:time2	0.4137078	0.3248364	-0.2229716	1.050387
N:(Intercept)	9.2480825	0.1432740	8.9672655	9.528899

```
Real Parameter p
```

```
Session:1
```

1	2
0.0900697	0.1302131

```
Real Parameter sigma
```

```
Session:1
```

```
0
```

```
NA
```

```
Real Parameter N
```

```
Session:1
```

```
10584.63
```

```
salmon.mt$results$real
```

	estimate	se	lcl	ucl	fixed
p g1 s1 t1	9.006970e-02	0.0202656	0.0574631	1.384607e-01	
p g1 s1 t2	1.302131e-01	0.0238270	0.0901817	1.844126e-01	
N g1 s1	1.058463e+04	1487.8477000	8053.3131000	1.393186e+04	
sigma g1 a0 s1 t0	0.000000e+00	0.0000000	0.0000000	0.000000e+00	Fixed
	note				
p g1 s1 t1					
p g1 s1 t2					
N g1 s1					
sigma g1 a0 s1 t0					

The estimated abundance is

```
get.real(salmon.mt, "N", se=TRUE)
```

```
    all.diff.index par.index estimate      se      lcl      ucl fixed note
N g1 s1           4       3 10584.63 1487.848 8053.313 13931.86
      group session
N g1 s1           1       1
```

We can fit a model where the capture probabilities are equal across both subsequent sampling events:

```
salmon.m0 <- mark( salmon.proc,
  model="LogitNormalMR",
  model.name="(p(.) sigma=0)",
  model.parameters=
    list(p = list(formula=~1),
         sigma=list(formula=~1, fixed=0),
         N = list(formula=~1)),
    initial=c(0,0,log(10000)) )
```

```
Output summary for LogitNormalMR model
Name : (p(.) sigma=0)

Npar : 2
-2lnL: 286.6689
AICc : 290.699

Beta
      estimate      se      lcl      ucl
p:(Intercept) -2.089332 0.1598152 -2.402570 -1.776095
N:(Intercept)  9.248113 0.1435710  8.966714  9.529512

Real Parameter p
Session:1
      1      2
0.110138 0.110138
```

```
Real Parameter sigma
```

```
Session:1
```

```
0
```

```
NA
```

```
Real Parameter N
```

```
Session:1
```

```
10584.95
```

```
salmon.m0$results$real
```

	estimate	se	lcl	ucl	fixed
p g1 s1 t1	0.110138	0.0156631	0.0829769	0.144786	
N g1 s1	10584.951000	1490.9778000	8049.0515000	13940.158000	
sigma g1 a0 s1 t0	0.000000	0.0000000	0.0000000	0.000000	Fixed
	note				
p g1 s1 t1					
N g1 s1					
sigma g1 a0 s1 t0					

The model comparison table is constructed in the usual way.

```
salmon.modset <- collect.models( type="LogitNormalMR")
salmon.modset
```

	model	npar	AICc	DeltaAICc	weight	Deviance
1	(p(.) sigma=0)	2	290.6990	0.0000000	0.5478212	286.6689
2	(p(t) sigma=0)	3	291.0828	0.3837426	0.4521788	285.0223

You CANNOT use the standard *model.average()* function due to the way *MARK* and *RMark* interact (groan)! Notice that the model averaged estimate is *N* is WRONG.

```
salmonavg.N.wrong <- model.average(salmon.modset, parameter="N", vcv=TRUE) # WRONG ANSWER
salmonavg.N.wrong
```

```
$estimates
      par.index estimate      se      lcl      ucl fixed note group session
N g1 s1        4 10384.81 1489.564 7839.803 13755.99                      1      1
```

```
$vcv.real
 4
4 2218800
```

The model averaged value of N is lower than each of the estimates which is impossible because it actually extracts the estimated UNMARKED population abundance from each subsequent capture event (groan).

You need to create a list structure to use *model.average.list()* function – see the help file on this function.

```
est.list <- function(modset){
  # Extract the estimate, AICc, and SE from the model sets
  # we need list with three vectors
  estimate <- plyr::lapply(modset[ names(modset) != "model.table"],
                           function(x){get.real(x, "N", se=TRUE)$estimate})
  weight   <- modset$model.table$weight
  se        <- plyr::lapply(modset[ names(modset) != "model.table"],
                           function(x){get.real(x, "N", se=TRUE)$se})
  list(estimate=estimate, weight=weight, se=se)
}
est.to.average <- est.list(salmon.modset)
est.to.average
```

```
$estimate
[1] 10584.95 10584.63
```

```
$weight
[1] 0.5478212 0.4521788
```

```
$se
[1] 1490.978 1487.848
```

This creates a list with 3 vectors corresponding to the estimates to be averaged, the weights for the estimates, and the SE of the individual model estimates.

Finally, we can model average using this data structure (groan):

```
salmonavg.N <- model.average(est.to.average, mata=TRUE)  # mata argument request ci's
salmonavg.N
```

```

$estimate
[1] 10584.81

$se
[1] 1489.563

$lcl
[1] 7665.316

$ucl
[1] 13504.3

[1] TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE

```

This procedure is not for the faint of heart!

- Capture histories are “imaginary” and are devices to getting the information into *MARK*. In particular the use of the *00* and *0.* histories to allow for marks to be added and removed is not intuitive.
- The total count of unmarked animals is specified in the *process.data()* function.
- Need to specify good starting values on the *beta* (logit or log scale depending on the parameter) scale (groan)!
- Cannot use the standard *model.average()* function and need to use the *model.average.list()* function (groan).

As a point of interest, the pairs of Petersen estimates formed at the second and third sampling event and by pooling the subsequent capture events are shown in Table 36.

Table 36: Mean of the Petersen estimator and Petersen estimator form by pooling all samples

Events	Estimate	SE
1 & 2	10,874	2292
1 & 3	9,721	1692
Average	10,297	????
Pooled	10,420	1340

12.3 Example of PNE method - TBA

12.4 Summary

13 Forward and reverse capture-recapture studies

13.1 Introduction

Consider a capture-recapture study of fish returning to spawn that consists of many distinct stocks with different spawning grounds.

In a forward capture-recapture recapture study, fish are captured at the entrance to the river, tagged and released. Then on one particular spawning ground, fish are examined for tags. But in many cases, biological samples are taken are also taken when fish are tagged, and genetic methods used to estimate the stock proportions.

This can be used in “reverse” capture-recapture study. If fish are fully enumerated on the spawning ground (e.g., at a fence), this is considered the “tagging” event. Then working backward, the biological samples are the “recapture” event.

This methodology is explained in more detail in Hamazaki and DeCovich (2014). The key assumptions for the genetic reverse-capture-recapture method (in addition to the usual for a Petersen estimator) include:

1. **Complete enumeration of stock on the spawning ground or at a weir.** A complete enumeration of the stock is needed to ensure that the “tagging” event is complete. This assumption could be violated by the stock spawning elsewhere (e.g., main stem), or not all spawning areas surveyed. Or multiple stocks with different genetic baselines could spawn in the same spawning grounds.
2. **The genetic baseline is complete and accurate.** The baseline needs to be complete in-order to confidently assign individuals to the “marked” (genetically distinct) population. If incomplete, individuals from genetically similar populations may be assigned to the “marked” population (see below) because it is the “closest” populations for assignment. This assumption can be relaxed if “not in baseline” is an acceptable assignment for stocks not in the baseline. The genetic baseline must also be accurate, e.g., fish the form the baseline really belong to the stock of interest and are not a mixture of multiple stocks.
3. **Marked population is genetically distinct.** The identified “marked” population needs to be genetically distinct from all other populations in the system so that individuals from genetically similar populations are not assigned to the “marked” population introducing bias in the estimated proportion of the “marked” population during the “second” sampling occasion at the entrance to the system.
4. **Marked population size.** While the size of the “marked population” theoretically has no bearing on the estimator, a small stock may have too few “recaptures” in the biological samples and so the estimator may have very poor precision.

5. **Equal en route mortality among different components of the run.** En route mortality in the reverse capture-recapture method acts like “immigration” in the forward capture-recapture methods and so leads to valid estimates at the entrance to the system. Homogeneous en route mortality may be violated when there are differential stock-specific harvest rates among stocks. This assumption can be assessed in the harvest through rigorous tissue collection of harvested individuals. The assumption may also be violated if stock have different distances between the entrance to the system and their spawning sites and so the en route mortality may also differ. One way to test this assumption is via the goodness of fit testing of equal recovery probabilities discussed earlier.
6. **Random sampling at the entry to the system.** Random sampling at the entry to the system could occur if, for example, fishwheels are sampling approximately proportional to the run and a constant fraction of the tagged fish have biological samples taken, e.g., every nth tagged fish is sampled.

13.2 Example - Run size of Yukon River Chinook

This is the 2011 data from Hamazaki and DeCovich (2014) taken from Table 2 of their paper.

Briefly, Chinook Salmon returning to Yukon River are counted in the lower river (Pilot Station) using hydroacoustic and sonar methods. At the same time, biological samples are taken throughout the run and the stock of origin was determined using GSI methods. In particular, in 2011, the proportion of stocks of Canadian origin was 0.35 (SE .030) based on a sample size of 251 fish of which 87 were fish of Canadian origin.

Fish continue their upward migration and the number of fish that enter Canadian waters was counted using sonar at the Eagle Border Station (51,271 (SE 135)). To this was added the estimated harvest of Canadian fish between Pilot Station and Eagle Border Station from harvest monitoring given a total of 66,225 (SE 1514) Canadian fish. These are considered to be the “tagged” fish.

Now consider running the experiment in reverse. We have 66,225 are marked at Eagle Border and swim “backwards” to Pilot Station. Here a biological sample of 251 fish were extracted, of which 87 were fish from Canada. This is the “recapture” sample. So, in reverse, we have $n_1 = 66,225$; $n_2 = 251$, $m_2 = 87$ giving a reverse Petersen estimate of

$$\widehat{N}_{\text{reverse}} = \frac{n_1 \times n_2}{m_2} = \frac{66,225 \times 251}{87} = 191,062$$

as reported in their paper. [Their estimate is slightly different because of rounding.]

The data are available in the *Petersen* package:

```
data(data_yukon_reverse)
data_yukon_reverse
```

```

cap_hist freq se
1      10 66138 1574.00000
2      11    87   7.53865
3      01   164   0.00000
                                comment
1 Escapement + harvest - genetic samples recaptured cdn fish
2                           Genetic sample that are cdn fish
3                           Genetic sample that are non-cdn fish

```

The estimates are obtained in the usual way:

```

yukon.fit <- Petersen::LP_fit(data_yukon_reverse,
                               p_model=~..time)
yukon.est <- Petersen::LP_est(yukon.fit, N_hat=~1)
yukon.est$summary

```

	N_hat_f	N_hat_rn	N_hat	N_hat_SE	N_hat_conf_level	N_hat_conf_method
1	~1	(Intercept)	191062.9	16546.88	0.95	logN
		N_hat_LCL	N_hat_UCL	p_model	name_model	cond.ll n parms nobs method
1	161234.7	226409.2	~..time	p: ~..time	-1812.538	2 66389 CondLik

The reported SE is too small because it does not account for the uncertainty in the n_1 (uncertainty in harvest and from sonar) and m_2 (because genetic assignment has some error). The *LP_est_adjust()* function can again be used. First, we get the adjustment factors for n_1 and m_2 based on the reported uncertainties:

```

n1.adjust.est <- 1
n1.adjust.se  <- data_yukon_reverse$se[ data_yukon_reverse$cap_hist=="10"] / sum(data_yukon_reverse$se)
cat("Estimated adjustment factors for n1 ", n1.adjust.est, n1.adjust.se, "\n")

```

Estimated adjustment factors for n1 1 0.02379872

```

m2.adjust.est <- 1
m2.adjust.se  <- data_yukon_reverse$se[ data_yukon_reverse$cap_hist=="11"] / sum(data_yukon_reverse$se)
cat("Estimated adjustment factors for m2 ", m2.adjust.est, m2.adjust.se, "\n")

```

Estimated adjustment factors for m2 1 0.03003446

And, now the adjustment factors are applied

```

set.seed(34543534)
yukon.adjust <- LP_est_adjust(
    yukon.est$summary$N_hat, yukon.est$summary$N_hat_SE,
    n1.adjust.est=n1.adjust.est, n1.adjust.se=n1.adjust.se,
    m2.adjust.est=m2.adjust.est, m2.adjust.se=m2.adjust.se
)
yukon.adjust$summary

```

	N_hat_un	N_hat_un_SE	N_hat_adj	N_hat_adj_SE	N_hat_adj_LCL	N_hat_adj_UCL
1	191062.9	16546.88	191996.4	18417.44	158396.5	230678

The final answer matches those reported in the paper ($\widehat{N}_{paper} = 191,155$ (SE 18,372)) where the authors used a bootstrapping procedure (similar to that used in the *LP_est_adjust()* function).

13.3 Example: Lower Fraser Coho - SPAS in reverse

This example was provided by Kaitlyn Dionne from the Department of Fisheries and Oceans, Canada and is discussed in Arbeider et al (2020) in more detail.

Arbeider et al (2020) proposed to estimate the run size of Lower Fraser River Coho (LFC) using a geographically stratified reverse-capture method. Briefly, as LFC coho swim upstream in the Fraser River, they are sampled near New Westminister, BC, which is downstream from several major rivers with large spawning populations. These sampled fish are assigned to the spawning population using genetic stock identification and other methods. These spawning populations are identified as the Chilliwack Hatchery (denoted C), the Lilloet River natural spawning population (denoted L), the Nicomen Slough population (denote N) and all other populations (denoted as O). Notice that the sampled fish at New Westminister are NOT physically tagged, and population assignment is through genetic stock identification and other measures.

The upstream migration extends over two months (September to October) and is divided into 3 temporal strata corresponding to *Early* (denoted $1E$), *Peak* (denoted $2P$) and *Late* (denoted $3L$). The digits 1, 2, 3 in front of the codes ensures that the temporal strata are sorted temporally, but this is merely a convenience and does not affect the results.

The spawning populations at C , L , and N are estimated by a variety of methods (see Arbeider, et al. 2020). Each of the population estimates also has an estimated SE (which will be ignored for now).

The data are available in the *Petersen* package:

```
data(data_lfc_reverse)
data_lfc_reverse
```

	cap_hist	freq	SE
1	C..1E	23	NA
2	C..2P	44	NA
3	C..3L	6	NA
4	C..0	66127	13039
5	L..1E	12	NA
6	L..2P	8	NA
7	L..3L	2	NA
8	L..0	14300	403
9	N..1E	1	NA
10	N..2P	13	NA
11	N..3L	3	NA
12	N..0	15274	461
13	O..1E	37	NA
14	O..2P	146	NA
15	O..3L	46	NA

This can be displayed in a matrix form as:

	s2			
s1	0	1E	2P	3L
0	0	37	146	46
C	66127	23	44	6
L	14300	12	8	2
N	15274	1	13	3

The first row (corresponding to $s1=0$) are the number of fish in the New Westminister sample assigned to stocks other than C , L , or N . The first column (corresponding to $s2=0$) are the estimated total escapement in each of C , L , N (less the fish removed for genetic identification which are trivially small). The matrix in the lower bottom of the above array shows the number of fish assigned from the New Westminister sample to the four (C , L , N , and O) geographic strata in each of the 3 time periods ($1E$, $2P$, or $3L$).

Notice that this data set up is going in REVERSE from total spawning escapement backwards to the genetic sample in New Westminister.

13.3.1 Fitting the SPAS model

The SPAS model is fit to the above data using the wrapper supplied with this package. We start with no pooling of rows or columns.

```
lfc..mod..1.fit <- Petersen::LP_SPAS_fit(data_lfc_reverse,
                                         model.id="LFC - entire matrix",
                                         row.pool.in=1:3, col.pool.in=1:3, quietly=TRUE)
```

The summary of the fit is:

```
lfc..mod..1.fit$summary
```

	p_model	name_model	cond.ll	n parms	nobs	method
1	Refer to row.pool/col.pool LFC - entire matrix		923265.5		15	96042 SPAS
	cond.factor					
1	2095.344					

The usual conditional likelihood is presented etc. Of more importance is the condition factor. This indicates how close to singularity the recovery matrix (in this case, the stock assignment of New Westminister fish to the geographic and temporal strata) is with larger values indicating that the matrix of “recoveries” is closer to singularity. Usually, this value should be 1000 or less to avoid numerical issues in the fit. Here the conditional factor is around 2000 which is a bit concerning but still likely to be acceptable.

We also fit a model (not shown) where the first two geographic strata (C and L) were pooled and give essentially the same abundance estimate.

We estimate the total LFC population:

```
lfc..mod..1.est <- Petersen::LP_SPAS_est(lfc..mod..1.fit)
lfc..mod..1.est$summary
```

	N_hat_f	N_hat_rn	N_hat	N_hat_SE	N_hat_conf_level	N_hat_conf_method	
1	NA	NA	289861.2	42516.79	0.95	Large sample	
	N_hat_LCL	N_hat_UCL			p_model	name_model	cond.ll
1	206529.8	373192.6	Refer to row.pool/col.pool	LFC - entire matrix	923265.5		
	n parms	nobs	method	cond.factor			
1	15	96042	SPAS	2095.344			

The reverse capture-recapture stratified-Petersen estimate of the total number of fish passing the New Westminister sampling station is 289,861 with a standard error of 42,517.

The pooled-Petersen estimator can be obtained by pooling all rows to a single row:

```
lfc..mod..PP.fit <- Petersen::LP_SPAS_fit(data_lfc_reverse, model.id="Pooled Petersen",
                                             row.pool.in=rep(1,3),
                                             col.pool.in=c(1,2,3), quietly=TRUE)
lfc..mod..PP.fit$summary
```

	p_model	name_model	cond.ll	n parms	nobs	method
1	Refer to row.pool/col.pool	Pooled Petersen	923258.6	13	96042	SPAS
	cond.factor					
1	1					

```
lfc..mod..PP.est <- Petersen::LP_SPAS_est(lfc..mod..PP.fit)
lfc..mod..PP.est$summary
```

	N_hat_f	N_hat_rn	N_hat	N_hat_SE	N_hat_conf_level	N_hat_conf_method
1	NA	NA	291716.4	22575.55	0.95	Large sample
	N_hat_LCL	N_hat_UCL			p_model	name_model
1	247469.1	335963.6	Refer to row.pool/col.pool	Pooled Petersen	923258.6	
	n parms	nobs	method	cond.factor		
1	13	96042	SPAS	1		

The capture-recapture reverse pooled-Petersen estimate of the total number of fish passing the New Westminister sampling station is 291,716 with a standard error of 22,576, which is virtually identical to the reversed stratified estimate but with a considerably reduced SE.

The above analysis ignored the uncertainty in the estimated escapement and so the reported standard errors for abundance are under reported. An approximate measure of this additional uncertainty for the pooled-Petersen reverse capture-recapture estimate could be obtained by bootstrapping, or using the *LP_est_adjust()* function.

```
# get the total of "n1" (reverse) and its approximate SE
temp <- data_lfc_reverse[ grep("0$", data_lfc_reverse$cap_hist),]
n1 <- sum(temp$freq)
n1.rse <- sqrt(sum(temp$SE^2))/n1

est.adj <- LP_est_adjust(mod..PP.est$summary$N_hat, lfc..mod..PP.est$summary$N_hat_SE, n1.adj)
```

Error: object 'mod..PP.est' not found

```
est.adj$summary
```

```
Error: object 'est.adj' not found
```

This shows that the uncertainty in the escapement in the terminal run would increase the SE of the abundance estimates by about 50%!

13.4 Example combining forward and reverse capture-recapture studies

TBA

13.5 Summary

The reverse capture-recapture method requires no mortality (either natural or harvest) between the location of the biological sample and the final spawning grounds. Otherwise, these sources of removal must be added to the “tagged” group when applied in reverse.

This is different from the forward capture-recapture experiment where mortality between the tagging and recapture events is allowable if it applies equally to tagged and untagged fish. When running the estimator in reverse, it would appear on first glance that any mortality should have similar effects to immigration where the abundance could still be estimated at the second sampling event. However, the key difference is that immigration in the forward direction applies only to untagged fish, but in the reverse direction mortality (now appearing to be immigration) is being applied to “tagged” fish.

14 Genetic capture-recapture methods using close kin studies

Genetic markers that uniquely identify individual can be used in place of marks or tags and pose no unique challenges. An interesting new application is the use of transgenerational marks where the set of animals sampled are different in the two events.

Single sample methods? Like the sequential capture-recapture methods taking one sample at a time? Rarefaction curves?

14.1 Transgenerational capture-recapture

14.1.1 Introduction

In these studies, The first sample is of adults is captured and a set of genetic markers (e.g., SNPs) are established for each captured animals. These adults mate and reproduce. A second sample of offspring is captured. The genetic information from these offspring is obtained. If the genetic markers are sufficiently diverse, then a parent-offspring-pair may be established in some cases, i.e., for some of the offspring, it can be established if the father and/or mother arose from the initial marked set (Figure 34).

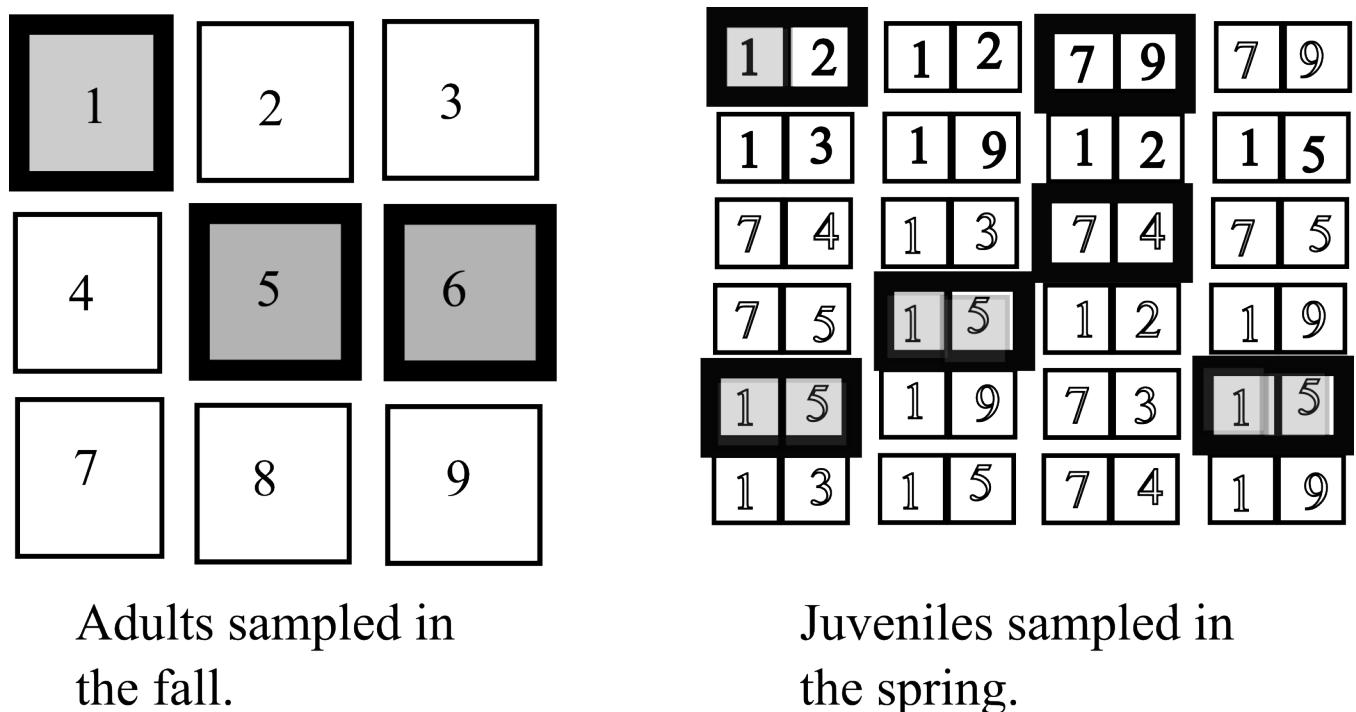


Figure 34: Schematic of transgenerational capture-recapture. Solid outline indicates a sampled fish. For juvenile fish, both the mother (left) and father (right) fish id are determined. Shaded areas indicate fish ids from genetic markers in juveniles that match genotyped adult fish. Notice that some fish (e.g. fish 8) did not produce offspring in the juvenile population. Mothers and fathers are not monogamous.

Notice that the set of animals sampled at the two sample events is different. Each juvenile sampled has two “genetic tags” (one for the father and one for the mother). Because a parental pair can give rise to many offspring and mother/father relationships may not be monogamous, multiple juveniles can have the “genetic tag” for a particular adults fish in the original samples.

Rawding et al (2014) used this method to estimate the number of returning Chinook salmon in the Ceweeman River, Washington State. The marks were the genotyped carcasses collected from the spawning area during the first sampling event. The second sampling event consisted of a collection of juveniles from a downstream migrant trap located below the spawning area. The parents that assigned to the juveniles through parentage analysis were considered the recaptures, which was a subset of the genotypes captured in the second sample.

Rosenbaum et al (2024) also used a similar method to estimate the number of returning Chinook salmon in the Chilkat River in Alaska State. Returning adult Chinook salmon were captured during freshwater migration in June and July using fishwheels and drift gillnets or in August using rod and reel, dip nets, short tangle nets, beach seines, and carcass surveys. Juvenile tissue samples were collected in the fall of 2021 using a stratified systematic sampling design, with samples collected continuously throughout September and October, across the mainstem and two of the three primary spawning tributaries using baited minnow traps.

In both examples, kinship relationships among adults and juveniles was determined using the parentage analysis program COLONY (Wang & Santure, 2009).

COLONY is a full probability pedigree reconstruction software that uses maximum likelihood to reconstruct full- and half-sibling family groups among juveniles and assigns parents to family groups. Additionally, COLONY infers genotypes of unsampled parents through information of sibling relationships among juveniles. Reconstructing unsampled parental genotypes allows for putative identification of the total number of reproductively successful adults, both sampled and unsampled.

14.1.2 Bailey Binomial model

In the first model, the genetically typed adults are treated as the marked sample, and each juvenile genetically sampled is treated as two (independent) draws from the population of adults. Because a parental pair can produce multiple offspring, the juvenile samples treated as sampling with replacement. These are the conditions suitable for a Bailey (1951, 1952) binomial estimator.

Rosenbaum et al (2024) provided the raw genetic matching data from their study. When processed, this gives rise to the summary data in Table 37:

Table 37: Summary data from transgenerational capture-recapture study on Chilkat River, Alaska.

Capture history	Total captures	Frequency
01	1	279
01	2	191

Capture history	Total captures	Frequency
01	3	86
01	4	21
01	5	9
01	6	6
01	7	5
01	9	1
10	0	434
11	1	96
11	2	31
11	3	11
11	4	5
11	5	1
11	6	1
11	7	2

A total of $M = 581$ adults were captured and genotyped (histories 10 and 11 in Table 37). Most were not “recaptured” by the father/mother genotyping of the juveniles, but 96 adults were “recaptured” in the juvenile sample once, 31 adults were captured 2x; etc.

A total of 682 juveniles were genotypes and each gives rise to 2 parental genotypes, some of which are matched, and 279 juvenile mother/father slots were not recaptures of the adults from previous fall. This gives $C = 1364$ (number of juvenile genotypes measured - two from each fish).

Finally, a total of $R = 236$ “recaptures” (including double, triple counting etc.) were found.

The (bias adjusted) Bailey Binomial estimate of adults abundance in the fall is

$$\widehat{N}_{BB} = \frac{M \times (C + 1)}{R + 1}$$

and found to be 3346 (SE 197) fish.

14.1.3 Hypergeometric model

The Bailey Binomial model assumes that on each draw (i.e., each juvenile) has an equal chance of selecting each adult parental genotype. However, a more fecund fish will likely give rise to more juveniles, and so the assumption of equal probability of sampling an adults fish is violated.

For this reason, the number of UNIQUE genotype in the juvenile sample can be used in the usual hypergeometric (Petersen or Chapman) estimator.

The value of M (number adults genotypes) is unchanged. The number of unique adults recaptured can be derived from Table 37 by treating multiple recaptures of the same adult genotype as single capture event. This give $R_{unique} = 147$.

Similarly, the number of unique adult genotypes seen for the first time is also determined from Table 37. For this study this gives $C_{unique} = 745$.

This gives an estimate of 2933 (SE 186) fish.

Both estimates are similar.

14.1.4 Caution on using simple sample methods.

It might be tempting to use single-sample methods based on the sample of juveniles alone such as species rarefaction curves, or methods developed for capture-recapture that allow for heterogeneity (such as Chao's estimate for the lower bound on the number of species). However, these methods will give you an estimate for the number of adults that successfully bred and had juveniles that survived to the time of sampling. This can be substantially less than the total number of adults that returned to spawn.

For example, Chao's simple estimator of

$$\widehat{N}_{Chao} = C_{unique} + \frac{f_1^2}{2f_2}$$

where f_i is the number of adult genetic codes captured i time in the juvenile sample, and C_{unique} is the number of unique adult genetic codes seen in the juvenile sample gives an estimate of 949 adults, which is substantially less than the other estimates, presumably because of the large number of returning adults who do not breed, nor had juveniles that survived to the time of sampling.

14.1.5 Limitations and Assumptions

The key assumptions needed to ensure that the Petersen (and related) estimators are unbiased are

- (1) there is no mark loss,
- (2) there are no marking effects,
- (3) the population is closed,
- (4) all marked and unmarked fish are correctly identified and enumerated, and
- (5) all fish in the population have an equal probability of capture in at least one sample event, or there is complete mixing of marked and unmarked individuals. There are additional, biologically unlikely, ways.

Genetic-based approaches meet the no tag-loss and no marking-effects assumptions, because genotypes were permanent markers and genetic marks were obtained from adults or from juveniles.

The closure assumption requires that within the study period there are no additions to the population through births or immigration and no deletions through death or emigration. In this studies, the closure assumption is likely violated by adults failing to produce offspring; however, the Petersen estimator is unbiased with respect to the population at time of tagging if mortality was random. Similarly, if the carcass collection was unbiased regarding the production of at least one offspring and the mortality (i.e., lack of spawning success) was equal for sampled and unsampled carcasses. There is likely no a priori reason to believe carcass sampling was not representative relative to the production of at least one offspring.

Regarding juveniles, careful study design is required to ensure that only juveniles from the adults population in the fall are sampled.

Good laboratory procedures and sufficient genetic markers are needed to ensure that correct identification and reporting of tagged or marked fish (assumption 4).

The last assumption (homogeneity of capture probabilities in at least one sample event) is the most difficult to ensure is being met. A representative carcass sampling design can be used to obtain a sample of parents. For tGMR, differences in individual parental reproductive success create unequal capture probabilities for parents in the second sampling event. When repeatedly sampling the same individual, the heterogeneity in capture probabilities can be partially alleviated by using only unique adults captures.

Equal mixing would imply that juveniles produced by the adults marked fish mix completely with the juveniles produced by unmarked adult fish. This could be violated by geographic separation of spawning adults, etc.

Rosenbaum et al (2024) concluded that the binomial tGMR estimate is less robust to assumption violations and more prone to biases as a result of following sampling with replacement. They caution tGMR users from relying on the binomial estimator. Instead, it is preferable to

identify an informative panel of genomic markers for the target population to confidently infer unsampled parents during parentage analysis so that the hypergeometric framework may be reliably utilized.

Heterogeneity can be (partially) alleviated through the use of covariates as presented early in this manuscript. However, covariates are readily available for the “marked” sample of adults, but not for the adults never seen before as identified by the juvenile sampling. It may be possible to identify sex by a marker, but other geographic or temporal covariates cannot be assigned to every unique adult seen. **This is a major limitation of the use of genetic methods in capture-recapture experiments.**

Rosenbaum et al (2024) also present a simulation program to examine the impact of various assumption violations on the estimates. Consult their paper for more details.

Rawding et al (2014) also conclude

The genetic methods we describe can be used with other anadromous salmonids that immigrate to the ocean shortly after emergence, such as Chum Salmon *Oncorhynchus keta* and Pink Salmon *O. gorbuscha*. Yet, additional considerations are needed to extend this application to anadromous salmonids with yearling life histories, such as spring Chinook Salmon, Coho Salmon *O. kisutch*, and Atlantic Salmon *Salmo salar*. Further, Atlantic Salmon and steelhead *O. mykiss* produce unique challenges because smolts may be the offspring of anadromous fish or resident fish or both. Using our methods, the Nc and Nb for steelhead would include the combined resident and anadromous spawning population, which may not meet manager needs, as the anadromous life history form is listed for protection under the ESA, while individuals exhibiting a resident life history are not listed.

The tGMR approach could also be extended to estimate adult abundance using only returning adults. Rather than the second sampling event consisting of juvenile captures, the second event would be a capture of returning adults. By aging the scales from returning adults, the adults could be assigned to the appropriate brood year based on scale ages, and tGMR could then be used to estimate spawner abundance for relevant brood years. Another variant of the genetic mark–recapture design is to estimate smolt abundance via “back-calculation” (Volkhardt et al. 2007). In this approach the smolts are the marks and the returning adults are the recaptures and captures.

15 Grand summary

The Petersen estimator is the simplest of capture-recapture studies but should not be analyzed using simplistic methods.

- Lessons from Petersen estimator area transferable to other experiments and models
- Use conditional MLE because it can deal with covariates in a unified fashion.

- Chapman correction factor only needed for small samples, but with small numbers of marks back, estimate will have poor precision.
- Confidence intervals can be found in many ways:
 - $\widehat{N} \pm 2se$ in large samples
 - Transform to $\log()$ scale, find ci, back transform in smallish samples

The key assumptions are:

- Population is closed
 - If death only, then estimates \widehat{N}_1
 - If births only, then estimates \widehat{N}_2
 - If births and deaths, then estimates total animals available
- No tag loss
 - Leads to positive bias in estimated population size
 - Need to double tag some or all of animals with permanent mark or second tag of same or different type
 - Need to adjust SE if only an estimate of tag loss is available.
- All tags seen and/or all tags reported
 - Leads to positive bias in estimate population size
 - Resample to look for lost tags; offer reward tags
 - Need to adjust SE if only estimate of tag reporting is available.
- No marking effect on catchability
 - Trap happiness leads to negative bias in population estimate
 - Trap shyness leads to positive bias in population estimate
 - Impossible to design study to adjust for problem as no control group possible
 - Perhaps use different trapping methods
- No marking effect on survival (e.g. no acute tagging mortality)
 - Leads to positive bias in population estimate
 - Impossible to design study to adjust for problem as no control group possible
- **Complete Mixing and/or homogeneity of catchability**
 - Pure heterogeneity leads to negative bias in population estimate
 - Mixed heterogeneity can lead to positive or negative bias in population estimate
 - Good study design to ensure that all of relevant population is sampled; gear should be non-selective
 - Regular, or Geographical (*SPAS*) or temporal stratification (*BTSPAS*) may be needed.

You are likely to unsatisfied with geographical stratification because of the many parameters that must be estimated and with sparse data, the high degree of singularity in the recapture matrix.

It is essential to recapture sufficient number of marked animals.

- Rule of thumb is that $rse \approx \frac{1}{\sqrt{marks\ returned}}$
- Precision can be improved by combining multiple Petersen estimates
- Possible to combine Petersen with other methods such as acoustic sampling; change in ratio; CPUE; etc (see me for details)

No amount of statistical wizardry can salvage a poorly designed study with insufficient marks back.

16 Miscellaneous notes

16.1 Speeding computation times

The computations in the examples in this monograph proceed quite quickly. However, there are times where the individual capture-history data structure may involve many thousands of individuals all with same capture-history.

Computation times can be considerably reduced by creating a condensed data set where the *freq* variable represents the number of animals with the same capture history. This speed up can be important when doing simulation studies.

```
data(data_NorthernPike)
```

Consider the NorthernPike example. This has rows for the 7805 individual fish in the capture-history array. This can be condensed to a similar format at the Rodli example:

```
# create condensed dataset
data_NorthernPike.condensed <- plyr::ddply(data_NorthernPike,
                                              c("cap_hist", "Sex"),
                                              plyr::summarize,
                                              freq=length(Sex))

data_NorthernPike.condensed
##   cap_hist Sex freq
## 1      01   F  524
## 2      01   M  459
## 3     10   F 3956
## 4     10   M 2709
```

```

## 5      11   F   89
## 6      11   M   68

```

Both data forms give the same estimates (as expected) but the difference in computation times is:

```

# Fit the various models
t.condensed.start <- Sys.time()
nop.condensed     <- Petersen::LP_fit(data_NorthernPike.condensed, p_model=~1+..time)
t.condensed.end   <- Sys.time()

t.full.start <- Sys.time()
nop.full      <- Petersen::LP_fit(data_NorthernPike,           p_model=~1+..time)
t.full.end    <- Sys.time()

cat("The two estimates \n")
## The two estimates
nop.condensed$summary
##      p_model      name_model      cond.ll n parms nobs   method
## 1 ~1 + ..time p: ~1 + ..time -3702.341      2 7805 cond ll
nop.full      $summary
##      p_model      name_model      cond.ll n parms nobs   method
## 1 ~1 + ..time p: ~1 + ..time -3702.341      2 7805 cond ll

cat("Time to run the condensed data structure: ", t.condensed.end - t.condensed.start, "secon"
## Time to run the condensed data structure: 0.08402109 seconds
cat("Time to run the full      data structure: ", t.full.end      - t.full.start,      "secon"
## Time to run the full      data structure: 1.009784 seconds

```

17 References

Akaike, H. (1973). Information Theory as an Extension of the Maximum Likelihood Principle. In Second International Symposium on Information Theory, edited by B. N. Petrov and F. Csaki, 267–281. Budapest: Akademiai Kiado.

Arbeider, M., Challenger, W., Dionne, K., Fisher, A., Noble, C., Parken, C., Ritchiel, L, and Robichaud, D. (2020) Estimating Aggregate Coho Salmon Terminal Run and Escapement to the Lower Fraser Management Unit. Report prepared for the Pacific Salmon Commission, dated 2020-02-21. Available at: <https://www.psc.org/fund-project/feasibility-of-estimating-aggregate-coho-salmon-escapement-to-the-lower-fraser-management-unit/>

- Arnason, A. N., and K. H. Mills. (1981). Bias and Loss of Precision Due to Tag Loss in Jolly-Seber Estimates for Mark-Recapture Experiments. Canadian Journal of Fisheries and Aquatic Sciences 38, 1077–1095.
- Arnason, A. N., C. W. Kirby, C. J. Schwarz, and J. R. Irvine. (1996). Computer analysis of data from stratified mark-recovery experiments for estimation of salmon escapements and other populations. Can. Tech. Rep. Fish. Aquat. Sci. 2106: vi+37 p.
- Bailey, N. T. J. (1951). On estimating the size of mobile populations from capture-recapture data. Biometrika 38, 293–306.
- Bailey, N. T. J. (1952). Improvements in the interpretation of recapture data. Journal of Animal Ecology 21, 120–127. <https://doi.org/10.2307/1913>
- Beverton, R. J. H., and Holt, S. J. (1957). On the dynamics of exploited fish populations. Fish. Invest. Minist. Mar. Fish. Minist. Agric., Fish. Food. (G.B.), Ser. 11, 19, 533 p.
- Bjorkstedt E. P. (2000). DARR (Darroch Analysis with Rank-Reduction): A method for analysis of stratified mark-recapture data from small populations, with application to estimating abundance of smolts from outmigrant trap data. Administrative Report SC-00-02, National Marine Fisheries Service. Available at: <http://swfsc.noaa.gov/publications/FED/00116.pdf>, Accessed 2009-07-28.
- Bonner Simon, J. and Schwarz Carl, J. (2011). Smoothing Population Size Estimates for Time-Stratified Mark Recapture Experiments Using Bayesian P-Splines. Biometrics, 67, 1498–1507.
- Bonner, S. J., & Holmberg, J. (2013). Mark-Recapture with Multiple, Non-Invasive Marks. Biometrics, 69, 766—775. <http://www.jstor.org/stable/24538143>
- Bonner, S. J. and Schwarz, C. J. (2023). BTSPAS: Bayesian Time Stratified Petersen Analysis System. R package version 2021.11.2.
- Bruesewitz, R., and K. Reeves. (2005). Estimating Northern Pike Abundance at Mille Lacs Lake, Minnesota. Fisheries Research Proposal. Minnesota Department of Natural Resource., Unpublished Material.
- Burnham, K. P., and Anderson, D. R. 2002. Model Selection and Multi-Model Inference: A Practical Information-Theoretic Approach. 2nd ed. New York: Springer.
- Chapman, D. H. (1951). Some Properties of the Hypergeometric Distribution with Applications to Zoological Censuses. Univ Calif Public Stat 1, 131–60.
- Chen, S. and Lloyd, C. (2000). A non-parametric approach to the analysis of two stage mark-recapture experiments. Biometrika 88, 649–663.
- Cochran, W. G. (1977). Sampling Techniques, 3rd Edition. 3rd ed. New York: Wiley.
- Cowen, Laura, and Schwarz, C. J. (2006). The Jolly-Seber Model with Tag Loss. Biometrics 62, 699–705. <https://doi.org/10.1111/j.1541-0420.2006.00523.x>.

- Darroch, J. N. (1961). The two-sample capture-recapture census when tagging and sampling are stratified. *Biometrika*, 48, 241–260. <https://www.jstor.org/stable/2332748>
- Goudie, I. B. J., and M. Goudie. (2007). Who Captures the Marks for the Petersen Estimator? *Journal of the Royal Statistical Society. Series A (Statistics in Society)*, 170, 825—839. <http://www.jstor.org/stable/4623202>
- Gulland, J. A. (1963). On the Analysis of Double-Tagging Experiments. Special Publication ICNAF No. 3, 228–29.
- Hamazaki, T. and DeCovich, N. (2014). Application of the Genetic Mark–Recapture Technique for Run Size Estimation of Yukon River Chinook Salmon. *North American Journal of Fisheries Management*, 34, 276–286. DOI: 10.1080/02755947.2013.869283
- Hyun, S.-Y., Reynolds.J.H., and Galbreath, P.F. (2012). Accounting for Tag Loss and Its Uncertainty in a Mark–Recapture Study with a Mixture of Single and Double Tags. *Transactions of the American Fisheries Society*, 141, 11–25. <http://dx.doi.org/10.1080/00028487.2011.639263>
- Huggins, R. M. 1989. On the Statistical Analysis of Capture Experiments. *Biometrika* 76: 133–40.
- Jackson, C. H. N. (1933). On the True Density of Tsetse Flies. *Journal of Animal Ecology* 2, 204–209.
- Junge, C. O. (1963). A Quantitative Evaluation of the Bias in Population Estimates Based on Selective Samples. I.C.N.A.F. Special Publication No. 4, 26–28.
- Laake J (2013). RMark: An R Interface for Analysis of Capture-Recapture Data with MARK. AFSC Processed Rep. 2013-01, Alaska Fish. Sci. Cent., NOAA, Natl. Mar. Fish. Serv., Seattle, WA. <https://apps.afsc.fisheries.noaa.gov/Publications/ProcRpt/PR2013-01.pdf>.
- Laplace, P. S. (1786). Sur Les Naissances, Les Mariages et Les Morts. *Histoire de l'Academie Royale Des Sciences, Annee 1783*, no. 693.
- Lincoln, F. C. (1930). Calculating Waterfowl Abundance on the Basis of Banding Returns. United States Department of Agriculture Circular No. 118, 1–4.
- Link, W. A. (2003). Nonidentifiability of Population Size from Capture-Recapture Data with Heterogeneous Detection Probabilities. *Biometrics* 59, 1123–1130.
- Link, W. A., and R. J. Barker. (2005). Modeling Association among Demographic Parameters in Analysis of Open Population Capture-Recapture Data. *Biometrics* 61, 46–54.
- McClintock, B.T., Conn, P.B., Alonso, R.S. and Crooks, K.R. (2013), Integrated modeling of bilateral photo-identification data in mark–recapture analyses. *Ecology*, 94, 1464–1471. <https://doi.org/10.1890/12-1613.1>
- McClintock, B. T. (2015) multimark: an R package for analysis of capture-recapture data consisting of multiple “noninvasive” marks. *Ecology and Evolution* 5, 4920–4931.

Mantyniemi, S. and Romakkaniemi, A. (2002). Bayesian mark-recapture estimation with an application to a salmonid smolt population. Canadian Journal of Fisheries and Aquatic Science 59, 1748–1758.

McClintock BT, White GC, Antolin MF, Tripp DW (2009) Estimating abundance using mark-resight when sampling is with replacement or the number of marked individuals is unknown. *Biometrics* 65, 237—246

McClintock, B.T., White, G.C. (2012) From NOREMARK to MARK: software for estimating demographic parameters using mark–resight methodology. *J Ornithol* 152 (Suppl 2), 641—650. <https://doi.org/10.1007/s10336-010-0524-x>

McDonald, T. L., S. C. Amstrup, and B. F. J. Manly. (2003). Tag Loss Can Bias Jolly-Seber Capture-Recapture Estimates. *Wildlife Society Bulletin* 31, 814–822.

Petersen, C. G. J. (1896). The Yearly Immigration of Young Plaice into the Limfjord from the German Sea, Etc. Report Danish Biological Station 6, 1–48.

Plante, N., L.-P Rivest, and G. Tremblay. (1988). Stratified Capture-Recapture Estimation of the Size of a Closed Population. *Biometrics* 54, 47–60. <https://www.jstor.org/stable/2533994>

Premarathna, W.A.L., Schwarz, C.J., Jones, T.S. (2018) Partial stratification in two-sample capture–recapture experiments. *Environmetrics*, 29:e2498. <https://doi.org/10.1002/env.2498>

Rawding, D.J., Sharpe, C.S. and Blankenship, S.M. (2014), Genetic-Based Estimates of Adult Chinook Salmon Spawner Abundance from Carcass Surveys and Juvenile Out-Migrant Traps. *Transactions of the American Fisheries Society*, 143, 55–67. <https://doi.org/10.1080/00028487.2013.829122>

Rajwani, K., and Schwarz, C.J. (1997). Adjusting for Missing Tags in Salmon Escapement Surveys. *Canadian Journal of Fisheries and Aquatic Sciences*, 54, 800–808.

Ricker, W. E. (1975). Computation and Interpretation of Biological Statistics of Fish Populations. *Bull Fish Res Board Can* 191.

Robson, D. S., and H. A. Regier. (1964). Sample Size in Petersen Mark-Recapture Experiments. *Transactions of the American Fish Society* 93, 215–26. [https://doi.org/10.1577/1548-8659\(1964\)93%5B215:SSIPME%5D2.0.CO;2](https://doi.org/10.1577/1548-8659(1964)93%5B215:SSIPME%5D2.0.CO;2)

Rosenbaum, S. W., May, S. A., Shedd, K. R., Cunningham, C. J., Peterson, R. L., Elliot, B. W., & McPhee, M. V. (2024). Reliability of trans-generational genetic mark–recapture (tGMR) for enumerating Pacific salmon. *Evolutionary Applications*, 17, e13647. <https://doi.org/10.1111/eva.13647>

Schwarz, C. J. (2023). SPAS: Stratified-Petersen Analysis System. R package version 2023.3.31.

Schwarz, C. J., Andrews, M., & Link, M. R. (1999). The Stratified Petersen Estimator with a Known Number of Unread Tags. *Biometrics*, 55, 1014—1021.

Schwarz, C. J. and Dempson, J. B. (1994). Mark-recapture estimation of a salmon smolt population. *Biometrics*, 50, 98–108.

Schwarz, C. J. and Taylor, C. G. (1998). The use of the stratified-Petersen estimator in fisheries management: estimating the number of pink salmon (*Oncorhynchus gorbuscha*) that spawn in the Fraser River. *Canadian Journal of Fisheries and Aquatic Sciences* 55, 281–297. <https://doi.org/10.1139/f97-238>

Seber, G. A. F. (1982). *The Estimation of Animal Abundance and Related Parameters*. 2nd ed. London: Griffin.

Seber, G. A. F., and R. Felton. (1981). Tag Loss and the Petersen Mark-Recapture Experiment. *Biometrika* 68, 211–19.

Seber, G.A. F. and M. R. Schofield (2023), Estimating Presence and Abundance of Closed Populations, *Statistics for Biology and Health*, Springer, New York. https://doi.org/10.1007/978-3-031-39834-6_1

Sprott, D. A. (1981). Maximum Likelihood Applied to a Capture-Recapture Model. *Biometrics* 37, 371–375.

Wang, J., & Santure, A. W. (2009). Parentage and Sibship inference from multilocus genotype data under polygamy. *Genetics*, 181(4), 1579—1594. <https://doi.org/10.1534/genetics.108.100214>

Weatherall, J. A. (1982). Analysis of Double Tagging Experiments. *Fishery Bulletin* 80, 687–701.

White, G.C. and K. P. Burnham. 1999.

Program MARK: Survival estimation from populations of marked animals. *Bird Study* 46 Supplement, 120–138.

Williams, B. K., J. D. Nichols, and M. J. Conroy. (2002). *Analysis and Management of Animal Populations*. New York: Academic Press.

Yee, T.W., Stoklosa, J., and Huggins, R.J. (2015). The VGAM Package for Capture-Recapture Data Using the Conditional Likelihood. *Journal of Statistical Software*, 65, 1–33. DOI: 10.18637/jss.v065.i05. URL <https://www.jstatsoft.org/article/view/v065i05/>.

papers by Rivest on the proper constant to add to the Petersen to make unbiased. See Canadian Journal of Statistics.

Morrison book

Paper on bayesian stratified petersen

paper on petersen with recognition errors Stevick, P., Palsbøll, P., Smith, T., Bravington, M., and Hammond, P.. (2011). Errors in identification using natural markings: rates, sources, and effects on capture-recapture estimates of abundance. *Canadian Journal of Fisheries and Aquatic Sciences*. 58. 1861-1870. DOI: 10.1139/f01-131.

18 Appendix A - The multinomial models for the Petersen study

As indicated earlier, the predominant paradigm currently used in capture-recapture studies is the multinomial model based on the observed capture histories. This formulation assume that neither n_1 nor n_2 nor m_2 are fixed quantities, but are all random variables.

There are two (asymptotically) equivalent multinomial formulations. The first is the complete multinomial where the abundance enters directly into the likelihood formulation. In this formulation an implicit assumption is that capture probabilities are known for animals never seen, i.e., are equal to the capture probabilities for animals that are seen in the study. In some cases, catchability is heterogeneous among fish, and the catchability may be related to a fixed covariates, such as fish length. Unfortunately, the fish length for fish never captured is unknown and so the capture probability for fish with history (00) cannot be modeled. Huggins (1989) showed that a conditional (upon fish been seen) multinomial model could be used in these circumstances. This chapter will develop these two formulations.

18.1 Complete multinomial model

The statistics for this formulation are the number with each capture history, n_{10}, n_{01}, n_{11} . Assuming no losses on capture, the probabilities of these histories and history (00) are

$$\begin{aligned} P(\{10\}) &= p_1 (1 - p_2) \\ P(\{01\}) &= (1 - p_1) p_2 \\ P(\{11\}) &= p_1 p_2 \\ P(\{00\}) &= (1 - p_1) (1 - p_2) \end{aligned}$$

The likelihood function is a multinomial distribution among these observed capture histories and probabilities:

$$L = \binom{N}{n_{00}, n_{01}, n_{10}, n_{11}} \times [p_1 (1 - p_2)]^{n_{10}} [(1 - p_1) p_2]^{n_{01}} [p_1 p_2]^{n_{11}} [(1 - p_1) (1 - p_2)]^{N - n_{10} - n_{01} - n_{11}}$$

The key difference between ordinary multinomial distributions and this likelihood is the presence of the unknown parameter N in the combinatorial term.

The estimators are found by maximizing the $\log(L)$ by taking first-derivatives, setting these to zero, and solving. The estimating equations for p_1 and p_2 are found to be:

- $(n_{11} + n_{10}) = Np_1$
- $(n_{11} + n_{01}) = Np_2$

Because the parameter N appears in factorial term, rather than taking first derivatives, the first difference is found, i.e., $\log(L(N)) - \log(L(N - 1)) = 0$ which leads to the estimating equation:

$$\left(\frac{N(1-p_1)(1-p_2)}{N - n_{10} - n_{01} - n_{11}} \right) = 0$$

Not surprisingly, this leads to the same estimators as seen earlier, namely:

$$\begin{aligned}\hat{N} &= \frac{(n_{11} + n_{10})(n_{11} + n_{01})}{n_{11}} = \frac{n_1 n_2}{m_2} \\ \hat{p}_1 &= \frac{(n_{11} + n_{10})}{\hat{N}} = \frac{n_{11}}{(n_{11} + n_{01})} = \frac{m_2}{n_2} \\ \hat{p}_2 &= \frac{(n_{11} + n_{01})}{\hat{N}} = \frac{n_{11}}{(n_{11} + n_{10})} = \frac{m_2}{n_1}\end{aligned}$$

Because it is possible for no marks to be recaptured ($m_2 = 0$), the same adjustments to the MLE can be made as found in the other estimators presented in Table 1.

The information matrix (the negative of the expected value of the second derivative matrix) is found by differentiating the first derivative or difference equations and replacing the observed statistics (the number of fish with each capture history) with its expected value. If the parameters are ordered by N , p_1 , and p_2 , the information matrix is:

$$\begin{bmatrix} \frac{1-(1-p_1)(1-p_2)}{N(1-p_1)(1-p_2)} & \frac{1}{(1-p_1)} & \frac{1}{(1-p_2)} \\ \frac{1}{(1-p_1)} & \frac{1}{p_1(1-p_1)} & 0 \\ \frac{1}{(1-p_2)} & 0 & \frac{1}{p_2(1-p_2)} \end{bmatrix}$$

The inverse of this matrix provides the variance-covariance matrix of the estimators:

$$\begin{bmatrix} \frac{N(1-p_1)(1-p_2)}{p_1 p_2} & \frac{-(1-p_1)(1-p_2)}{p_1(1-p_1)} & \frac{-(1-p_1)(1-p_2)}{p_1(1-p_2)} \\ \frac{-(1-p_1)(1-p_2)}{p_1(1-p_2)} & \frac{p_2}{N p_2} & \frac{p_1}{N p_1} \\ \frac{-(1-p_1)(1-p_2)}{p_1} & \frac{(1-p_1)(1-p_2)}{N} & \frac{p_2(1-p_2)}{N p_1} \end{bmatrix}$$

The precision of the estimated abundance is

$$se(\hat{N}) = \sqrt{N \frac{(1-p_1)}{p_1} \frac{(1-p_2)}{p_2}}$$

If the capture probabilities are small, then $\frac{1-p_i}{p_i}$ is large, and the precision is poor. An estimated SE is found by replacing the unknown parameters by their estimators to give:

$$\widehat{se}(\hat{N}) = \sqrt{\frac{n_1 n_2}{m_2} \frac{(n_2 - m_2)}{m_2} \frac{(n_1 - m_2)}{m_2}}$$

which again is very similar to the results presented in Table 1. Again, similar sorts of adjustments can be made (adding 1 to various terms) to avoid problems with zero counts.

18.2 Conditional multinomial model

The Huggins (1989) conditional multinomial model is most useful when the capture probabilities are modeled as functions of individual covariates and is not needed for the basic Petersen estimator. However, this development that follows will show how the basic conditional model is developed.

Now only animals that are seen at least once are used in the likelihood. The statistics for this formulation are the number with each capture history, n_{10}, n_{01}, n_{11} . Assuming no losses on capture, the conditional probability of each of these histories is found by normalizing the probabilities found in the previous section, by the total probability of been seen in the study.

$$P(\{10\}) = \frac{p_1(1-p_2)}{1-(1-p_1)(1-p_2)}$$

$$P(\{01\}) = \frac{(1-p_1)p_2}{1-(1-p_1)(1-p_2)}$$

$$P(\{11\}) = \frac{p_1p_2}{1-(1-p_1)(1-p_2)}$$

Notice that the probability of history (00) is not needed.

The conditional likelihood function is a multinomial distribution among these observed capture histories and probabilities:

$$L = \binom{n_{obs}}{n_{01}, n_{10}, n_{11}} \times \left[\frac{p_1(1-p_2)}{1-(1-p_1)(1-p_2)} \right]^{n_{10}} \left[\frac{(1-p_1)p_2}{1-(1-p_1)(1-p_2)} \right]^{n_{01}} \left[\frac{p_1p_2}{1-(1-p_1)(1-p_2)} \right]^{n_{11}}$$

where $n_{obs} = n_{01} + n_{10} + n_{11}$. Now no unknown parameter appears in the combinatorial term and the population abundance is not modeled..

The estimators are found by maximizing the $\log(L)$ by taking first-derivatives, setting these to zero, and solving. The conditional maximum likelihood estimates for p_1 and p_2 are found to be the same as in the complete multinomial case:

- $\hat{p}_1 = \frac{n_{11}}{(n_{11}+n_{01})}$
- $\hat{p}_2 = \frac{n_{11}}{(n_{11}+n_{10})}$

The information matrix (the negative of the expected value of the second derivative matrix) is found by differentiating the first derivative or difference equations and replacing the observed statistics (the number of fish with each capture history) with its expected value. The inverse of this gives the covariance matrix of the estimates. If the parameters are ordered by p_1 , and p_2 , the covariance matrix is:

$$Var_{cond} \begin{bmatrix} \hat{p}_1^{cond} \\ \hat{p}_1^{cond} \end{bmatrix} = \begin{bmatrix} \frac{p_1(1-p_1)}{n_{obs}p_2/(1-(1-p_1)(1-p_2))} & \frac{(1-p_1)(1-p_2)}{n_{obs}/(1-(1-p_1)(1-p_2))} \\ \frac{(1-p_1)(1-p_2)}{n_{obs}/(1-(1-p_1)(1-p_2))} & \frac{p_2(1-p_2)}{n_{obs}p_1/(1-(1-p_1)(1-p_2))} \end{bmatrix} \quad (3)$$

The variances of the capture probabilities are very similar to those in the complete multinomial models being of the form $\frac{p_i(1-p_i)}{Expected\ number\ of\ fish\ captured\ at\ the\ other\ occasion}$. There is asymptotically no loss in information about the capture probabilities by using the conditional model – again this is not surprising as the unknown number of fish never seen can't provide information about the capture probabilities.

The estimated variances are found by substituting in the estimated capture probabilities and reduce to a surprisingly simple form:

$$\widehat{\text{var}}_{cond} \begin{bmatrix} \hat{p}_1^{cond} \\ \hat{p}_1^{cond} \end{bmatrix} = \begin{bmatrix} \frac{n_{11}n_{01}}{\frac{(n_{01}+n_{11})^3}{n_{11}n_{10}n_{01}}} & \frac{n_{11}n_{10}n_{01}}{\frac{(n_{01}+n_{11})^2(n_{10}+n_{11})^2}{(n_{10}+n_{11})^3}} \\ \frac{(n_{01}+n_{11})^2(n_{10}+n_{11})^2}{(n_{10}+n_{11})^3} & \frac{n_{11}n_{10}n_{01}}{\frac{(n_{11}n_{10})^2}{(n_{10}+n_{11})^3}} \end{bmatrix}$$

Huggins (1989) showed that an estimator for the abundance can be obtained as:

$$\widehat{N}_{cond} = \sum_{\text{obs animals}} \frac{1}{\hat{p}_a^*}$$

where \hat{p}_a^* is the probability of seeing animal a in the study. In this simple Petersen study all animals have the same probability of being seen in the study, namely $1 - (1 - p_1)(1 - p_2)$ and the estimated abundance reduces to:

$$\widehat{N}_{cond} = \frac{n_{obs}}{1 - (1 - \hat{p}_1)(1 - \hat{p}_2)} = \frac{(n_{10} + n_{11})(n_{01} + n_{11})}{n_{11}} = \frac{n_1 n_2}{m_2}$$

which is the familiar Petersen estimator.

Huggins (1989) also gave an expression for the estimated variance of \widehat{N} :

$$\widehat{\text{var}}(\widehat{N}_{cond}) = \sum_{\text{obs animals}} \frac{1 - \hat{p}_a^*}{(\hat{p}_a^*)^2} + \widehat{D}^T \widehat{I}^{-1} \widehat{D}$$

where \widehat{D} is the partial derivative of the estimator \widehat{N} with respect to the catchabilities p_1 and p_2 , and \widehat{I}^{-1} is the estimated covariance matrix of the estimated catchabilities.

Long and tedious algebra reduces to the familiar form:

$$\widehat{\text{var}}(\widehat{N}_{cond}) = \frac{n_{01}n_{10}(n_{01} + n_{11})(n_{10} + n_{11})}{n_{11}^3} = \frac{n_1 n_2 (n_1 - m_2)(n_2 - m_2)}{m_2^3}$$

The conditional multinomial estimator is fully efficient – a conclusion echoed by Link and Barker (2005) who showed that there is no loss in information in using a conditional multinomial model to estimate abundance in the more general open-population capture-recapture setting.

For these reasons, we will ONLY use the conditional multinomial model in the *Petersen* package following the work of Yee et al (2015).

19 Appendix B - Using the *VGAM* package

Yee et al. (2015) developed the *VGAM* package which includes functions for fitting closed-population capture-recapture models, of which the Petersen-type studies are special cases. We will illustrate the use of this package with the Rödli Tarn data.

19.1 Data structure

The data structure to use the *VGAM* functions is slightly different from the data structure used by the *Petersen* package.

A data.frame is constructed with

- Two variables (traditionally called $t1$ and $t2$) for the capture history which must be numeric columns with 0 representing not captured and 1 representing captured. The *Petersen* package includes the `split_cap_hist()` function that can split the capture history represented by a single string.
- Each animal must be represented by its own record. Hence, grouped capture histories with a *freq* variable must be expanded to individual records.

See details in the examples below.

Yee et al. (2015) provides supplemental materials with code examples from which the following were derived.

19.2 Example Rodli Tarn

This was analyzed using the *Petersen* package in Section 3.5.1.

In this example, we first demonstrate how to split the capture histories and expand capture histories to individual records required for the *VGAM* functions.

```
library(VGAM)

data(data_rodli)

# split the capture histories
rodli.vgam <- cbind(data_rodli,
                      Petersen::split_cap_hist(data_rodli$cap_hist,
                                                make.numeric=TRUE))

# need to expand capture history by frequency
rodli.vgam.expand <- plyr::adply(rodli.vgam, 1, function(x){
```

```

x[ rep(1, x$freq),]
})

head(rodli.vgam.expand)
##   cap_hist freq t1 t2
## 1      11    57  1  1
## 2      11    57  1  1
## 3      11    57  1  1
## 4      11    57  1  1
## 5      11    57  1  1
## 6      11    57  1  1

```

Then the traditional Petersen estimator is the Mt model (in the parlance of the closed-population capture-recapture models).

```

rodli.vgam.m.t <- vglm(cbind(t1,t2) ~ 1,
                        posbernoulli.t, data = rodli.vgam.expand)
summary(rodli.vgam.m.t)
##
## Call:
## vglm(formula = cbind(t1, t2) ~ 1, family = posbernoulli.t, data = rodli.vgam.expand)
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept):1 -0.74444    0.16086 -4.628 3.7e-06 ***
## (Intercept):2  0.09181    0.19177  0.479   0.632
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Names of linear predictors: logitlink(E[t1]), logitlink(E[t2])
##
## Log-likelihood: -233.9047 on 456 degrees of freedom
##
## Number of Fisher scoring iterations: 4
##
## No Hauck-Donner effect found in any of the estimates
##
##
## Estimate of N:  338.474
##
## Std. Error of N: 25.496
##
```

```
## Approximate 95 percent confidence interval for N:
## 288.5 388.45
```

19.3 Example Rodli Tarn - Chapman correction

This was analyzed using the *Petersen* package in Section 3.5.2.

```
library(VGAM)

data(data_rodli)

# add extra animal that is tagged and recaptured
rodli.chapman <- plyr::rbind.fill(data_rodli,
                                    data.frame(cap_hist="11", freq=1, comment="Added for Chapman"))

# split the capture history
rodli.vgam.chapman <- cbind(rodli.chapman,
                             Petersen::split_cap_hist(rodli.chapman$cap_hist,
                                                       make.numeric=TRUE))

# need to expand capture history by frequency
rodli.vgam.chapman.expand <- plyr::adply(rodli.vgam.chapman, 1, function(x){
  x[ rep(1, x$freq),]
})

head(rodli.vgam.chapman.expand)
##   cap_hist freq comment t1 t2
## 1      11    57     <NA>  1  1
## 2      11    57     <NA>  1  1
## 3      11    57     <NA>  1  1
## 4      11    57     <NA>  1  1
## 5      11    57     <NA>  1  1
## 6      11    57     <NA>  1  1

# fit the Mt model with the Chapman correction
rodli.vgam.M.t.chapman <- vglm(cbind(t1,t2) ~ 1,
                                 posbernoulli.t, data = rodli.vgam.chapman.expand)
summary(rodli.vgam.M.t.chapman)
##
## Call:
## vglm(formula = cbind(t1, t2) ~ 1, family = posbernoulli.t, data = rodli.vgam.chapman.expand)
```

```

## 
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept):1 -0.7270    0.1599 -4.546 5.46e-06 ***
## (Intercept):2   0.1092    0.1910  0.572   0.567
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## 
## Names of linear predictors: logitlink(E[t1]), logitlink(E[t2])
## 
## Log-likelihood: -235.2889 on 458 degrees of freedom
## 
## Number of Fisher scoring iterations: 4
## 
## No Hauck-Donner effect found in any of the estimates
## 
## 
## Estimate of N:  337.586
## 
## Std. Error of N:  25.024
## 
## Approximate 95 percent confidence interval for N:
## 288.54 386.63

```

19.4 Example - Rodli Tarn - Equal capture probabilities

This was analyzed by the *Petersen* package in Section 3.5.3.

```

library(VGAM)

data(data_rodli)

# split the capture history
rodli.vgam <- cbind(data_rodli,
                      Petersen::split_cap_hist(data_rodli$cap_hist,
                      make.numeric=TRUE))

# need to expand capture history by frequency
rodli.vgam.expand <- plyr::adply(rodli.vgam, 1, function(x){
  x[ rep(1, x$freq),]
})

```

```

# fit the model
rodli.vgam.m.0 <- vglm(cbind(t1,t2) ~ 1,
                         posbernoulli.t(parallel = TRUE ~ 1), data = rodli.vgam.expand)
summary(rodli.vgam.m.0)
##
## Call:
## vglm(formula = cbind(t1, t2) ~ 1, family = posbernoulli.t(parallel = TRUE ~
## 1), data = rodli.vgam.expand)
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept) -0.4113    0.1528 -2.691  0.00712 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Names of linear predictors: logitlink(E[t1]), logitlink(E[t2])
##
## Log-likelihood: -247.7207 on 457 degrees of freedom
##
## Number of Fisher scoring iterations: 4
##
## No Hauck-Donner effect found in any of the estimates
##
##
## Estimate of N:  358.754
##
## Std. Error of N:  28.577
##
## Approximate 95 percent confidence interval for N:
## 302.74 414.76

# aic table
library(AICcmodavg)

AICcmodavg::aictab(list(rodli.vgam.m.0,
                        rodli.vgam.m.t))
##
## Model selection based on AICc:
##
##      K   AICc Delta_AICc AICcWt Cum.Wt       LL
## Mod2 2 471.86        0.0      1      1 -233.90
## Mod1 1 497.46       25.6      0      1 -247.72

```

19.5 Example - Northern Pike - Stratification by Sex

This was analyzed by the *Petersen* package in Section 6.2.

```
library(VGAM)

data(data_NorthernPike)
# split the capture history
NorthernPike.vgam <- cbind(data_NorthernPike,
                           Petersen::split_cap_hist(data_NorthernPike$cap_hist,
                           make.numeric=TRUE))

nop.vgam.time      <- vglm(cbind(t1,t2) ~ 1,
                           posbernoulli.t,
                           data = NorthernPike.vgam)
nop.vgam.sex.time  <- vglm(cbind(t1,t2) ~ Sex,
                           posbernoulli.t,
                           data = NorthernPike.vgam)
nop.vgam.sex.p.time <- vglm(cbind(t1,t2) ~ Sex,
                           posbernoulli.t(parallel = TRUE ~ Sex),
                           data = NorthernPike.vgam)

# aic table
library(AICcmodavg)

AICcmodavg::aictab(list(nop.vgam.time,
                        nop.vgam.sex.time,
                        nop.vgam.sex.p.time))

##
## Model selection based on AICc:
##
##      K      AICc Delta_AICc AICcWt Cum.Wt      LL
## Mod1 2  7408.68      0.00   0.71   0.71 -3702.34
## Mod2 3  7410.46      1.78   0.29   1.00 -3702.23
## Mod3 2 12143.55    4734.86   0.00   1.00 -6069.77
```

It is not possible to fit a model where the capture probabilities are equal at the first or second sampling event using the VGAM package.

19.6 Example - Northern Pike - Stratification by Length Class

```
library(VGAM)

data(data_NorthernPike)
# split the capture history
NorthernPike.vgam <- cbind(data_NorthernPike,
                           Petersen::split_cap_hist(data_NorthernPike$cap_hist,
                           make.numeric=TRUE))

# create the length classes
NorthernPike.vgam$length.class <- car::recode(NorthernPike.vgam$length,
                                                 " lo:20='00-20';
                                                 20:25='20-25';
                                                 25:30='25-30';
                                                 30:35='30-35';
                                                 35:hi='35+' ")

nop.vgam.time           <- vglm(cbind(t1,t2) ~ 1,
                                 posbernoulli.t,
                                 data = NorthernPike.vgam)
nop.vgam.length.class.time <- vglm(cbind(t1,t2) ~ length.class,
                                      posbernoulli.t,
                                      data = NorthernPike.vgam)
nop.vgam.length.class.p.time <- vglm(cbind(t1,t2) ~ length.class,
                                       posbernoulli.t(parallel = TRUE ~ length.class),
                                       data = NorthernPike.vgam)

# aic table
library(AICcmodavg)

AICcmodavg::aictab(list(nop.vgam.time,
                        nop.vgam.length.class.time,
                        nop.vgam.length.class.p.time))
## 
## Model selection based on AICc:
## 
##      K      AICc Delta_AICc AICcWt Cum.Wt      LL
## Mod2 6  7368.04     0.00     1     1 -3678.01
## Mod1 2  7408.68    40.64     0     1 -3702.34
## Mod3 5 12101.12   4733.09     0     1 -6045.56
```

It is not possible to fit a model where the capture probabilities are equal at the first or second sampling event using the VGAM package.

20 Appendix C. Using RMark/MARK

MARK (White and Burnham, 1999) is a Windows-based program for the analysis of capture-recapture experiments that can analyze many different types of studies *RMark* (Laake, 2013) is an *R* front-end that allows for a formula-type specification for models (rather than the graphical interface used in *MARK*), but then calls *MARK* for actual computation.s

In this section, we will demonstrate how to use *RMark/MARK* for the analysis of basic capture-recapture experiments.

20.1 Data structure

The data structure to use the *RMark/MARK* functions is similar to the data structure used by the *Petersen* package.

A data.frame is constructed with

- variable *ch* representing the capture history (e.g., 01, 10, 11)
- a variable *freq* representing the number of animals with that history
- other variables indicating groups and covariates.

See details in the examples below.

20.2 Example Rodli Tarn

This was analyzed using the *Petersen* package in Section 3.5.1.

We first get the Rodli data and then change the capture history vector variable name:

```
library(RMark)

data(data_rodli)
data_rodli.mark <- plyr::rename(data_rodli, c("cap_hist"="ch"))
data_rodli.mark
##   ch freq
## 1 11   57
## 2 10   52
## 3 01  120
```

We will used the Huggins-conditional closed-population likelihood models.

```
# We will be fitting some closed population models the "Closed" models, so we need to know
# the parameters of the model

setup.parameters("Huggins", check=TRUE) # returns a vector of parameter names (case sensitive)
## [1] "p" "c"

# Notice that there is NO parameter for the population size in the Huggins model

rodli.rmark.mt <- mark( data_rodli.mark, invisible=TRUE,
                         model="Huggins",
                         model.name="model ~..time",
                         model.parameters=list(p=list(formula=~time, share=TRUE)))
)

## 
## Output summary for Huggins model
## Name : model ~..time
##
## Npar : 2
## -2lnL: 467.8095
## AICc : 471.8358
##
## Beta
##           estimate      se      lcl      ucl
## p:(Intercept) -0.7444405 0.1608639 -1.0597337 -0.4291472
## p:time2        0.8362480 0.1660244  0.5108402  1.1616559
##
## 
## 
## Real Parameter p
##       1      2
## 0.3220339 0.5229358
##
## 
## 
## Real Parameter c
##       2
## 0.5229358
```

This gives much output, but we extract the relevant parts:

```
summary(rodli.rmark.mt, se=TRUE)
## Output summary for Huggins model
```

```

## Name : model ~..time
##
## Npar : 2
## -2lnL: 467.8095
## AICc : 471.8358
##
## Beta
##           estimate      se      lcl      ucl
## p:(Intercept) -0.7444405 0.1608639 -1.0597337 -0.4291472
## p:time2        0.8362480 0.1660244  0.5108402  1.1616559
##
## Real Parameter p
##           all.diff.index par.index estimate      se      lcl      ucl fixed
## p g1 t1             1          1 0.3220339 0.0351211 0.2573604 0.3943300
## p g1 t2             2          2 0.5229358 0.0478409 0.4294597 0.6148324
##
## Real Parameter c
##           all.diff.index par.index estimate      se      lcl      ucl fixed
## c g1 t2             3          2 0.5229358 0.0478409 0.4294597 0.6148324
rodli.rmark.mt$results$real
##           estimate      se      lcl      ucl fixed note
## p g1 t1 0.3220339 0.0351211 0.2573604 0.3943300
## p g1 t2 0.5229358 0.0478409 0.4294597 0.6148324

```

As before, the estimate of the population size is obtained using a Horvitz-Thompson-like estimator and is not part of the likelihood. These are known as derived parameters in *RMark/MARK*:

```

rodli.rmark.mt$results$derived
## $`N Population Size`
##   estimate      se      lcl      ucl
## 1 338.4737 25.49646 298.7707 400.7696

```

20.3 Example Rodli Tarn - Chapman correction

This was analyzed using the *Petersen* package in Section 3.5.2.

```

library(RMark)

data(data_rodli)
data_rodli.mark <- plyr::rename(data_rodli, c("cap_hist"="ch"))

# add extra animal that is tagged and recaptured
data_rodli.chapman.mark <- plyr::rbind.fill(data_rodli.mark,
                                              data.frame(ch="11", freq=1, comment="Added for Chapman"))

# fit the Mt model with the Chapman correction
rodli.rmark.mt.chapman <- mark( data_rodli.chapman.mark, invisible=TRUE,
                                 model="Huggins",
                                 model.name="model ~..time",
                                 model.parameters=list(p=list(formula=~time, share=TRUE))
                               )
## 
## Output summary for Huggins model
## Name : model ~..time
##
## Npar : 2
## -2lnL: 470.5777
## AICc : 474.604
##
## Beta
##           estimate       se      lcl      ucl
## p:(Intercept) -0.7270487 0.1599210 -1.0404938 -0.4136036
## p:time2        0.8362480 0.1660244  0.5108402  1.1616558
##
## 
## 
## Real Parameter p
##      1      2
## 0.3258427 0.5272727
##
## 
## 
## Real Parameter c
##      2
## 0.5272727
summary(rodli.rmark.mt.chapman, se=TRUE)
## Output summary for Huggins model
## Name : model ~..time
##
## Npar : 2

```

```

## -2lnL: 470.5777
## AICc : 474.604
##
## Beta
##           estimate      se      lcl      ucl
## p:(Intercept) -0.7270487 0.1599210 -1.0404938 -0.4136036
## p:time2        0.8362480 0.1660244  0.5108402  1.1616558
##
##
## Real Parameter p
##           all.diff.index par.index estimate      se      lcl      ucl fixed
## p g1 t1             1          1 0.3258427 0.0351297 0.2610547 0.3980483
## p g1 t2             2          2 0.5272727 0.0476022 0.4341067 0.6185773
##
##
## Real Parameter c
##           all.diff.index par.index estimate      se      lcl      ucl fixed
## c g1 t2             3          2 0.5272727 0.0476022 0.4341067 0.6185773
rodli.rmark.mt.chapman$results$real
##           estimate      se      lcl      ucl fixed note
## p g1 t1 0.3258427 0.0351297 0.2610547 0.3980483
## p g1 t2 0.5272727 0.0476022 0.4341067 0.6185773
rodli.rmark.mt.chapman$results$derived
## $`N Population Size`
##   estimate      se      lcl      ucl
## 1 337.5862 25.02399 298.6072 398.7109

```

We need to remove the Chapman model because the data is different when we compare model using AIC (later)"

```

rm(rodli.rmark.mt.chapman)
cleanup(ask=FALSE)

```

20.4 Example - Rodli Tarn - Equal capture probabilities

This was analyzed by the *Petersen* package in Section 3.5.3.

```

rodli.rmark.m0 <- mark( data_rodli.mark, invisible=TRUE,
                         model="Huggins",
                         model.name="model ~1",
                         model.parameters=list(p=list(formula=~1, share=TRUE))

```

```

        )
## 
## Output summary for Huggins model
## Name : model ~1
##
## Npar : 1
## -2lnL: 495.4414
## AICc : 497.4501
##
## Beta
##           estimate      se      lcl      ucl
## p:(Intercept) -0.411296 0.1528326 -0.710848 -0.111744
##
## 
## 
## Real Parameter p
##       1      2
## 0.3986014 0.3986014
##
## 
## 
## Real Parameter c
##       2
## 0.3986014

```

We now can do the usual AIC model averaging on the two models

```

# Collect results and get the AIC tables
rodli.results <- collect.models( type="Huggins")
rodli.results
##          model npar     AICc DeltaAICc      weight Deviance
## 2 model ~..time    2 471.8358  0.00000 9.999973e-01 2037.917
## 1     model ~1     1 497.4501 25.61429 2.741112e-06 2065.549

```

20.5 Example - Northern Pike - Stratification by Sex

This was analyzed by the *Petersen* package in Section 6.2.

We create the proper data structures:

```

data(data_NorthernPike)
data_nop.mark <- plyr::rename(data_NorthernPike, c("cap_hist"="ch"))
head(data_nop.mark)

```

```

##   ch length Sex freq
## 1 01  23.20   M   1
## 2 01  28.89   F   1
## 3 01  25.20   M   1
## 4 01  22.20   M   1
## 5 01  25.00   M   1
## 6 01  24.70   M   1

# We will be fitting some closed population models the "Closed" models, so we need to know
# the parameters of the model

setup.parameters("Huggins", check=TRUE) # returns a vector of parameter names (case sensitive)
## [1] "p" "c"

# Notice that there is NO parameter for the population size in the Huggins model

nop.sex.time <- mark( data_nop.mark, invisible=TRUE,
                      model="Huggins",
                      groups=c("Sex"), # note name must match case and that used in convert.inp
                      model.name="model sex*..time",
                      model.parameters=list(p=list(formula=~Sex*time, share=TRUE)))
)

##
## Output summary for Huggins model
## Name : model sex*..time
##
## Npar : 4
## -2lnL: 7391.653
## AICc : 7399.655
##
## Beta
##           estimate      se      lcl      ucl
## p:(Intercept) -1.7728550 0.1146483 -1.9975658 -1.5481443
## p:SexM        -0.1366882 0.1732871 -0.4763308  0.2029545
## p:time2       -2.0214974 0.0464884 -2.1126147 -1.9303800
## p:SexM:time2   0.2462126 0.0686218  0.1117138  0.3807114
##
## 
## Real Parameter p
##           1      2
## Group:SexF 0.1451876 0.0220025
## Group:SexM 0.1290322 0.0244868

```

```

## 
## 
## Real Parameter c
##          2
## Group:SexF 0.0220025
## Group:SexM 0.0244868

summary(nop.sex.time, se=TRUE)
## Output summary for Huggins model
## Name : model sex*..time
##
## Npar : 4
## -2lnL: 7391.653
## AICc : 7399.655
##
## Beta
##           estimate      se      lcl      ucl
## p:(Intercept) -1.7728550 0.1146483 -1.9975658 -1.5481443
## p:SexM         -0.1366882 0.1732871 -0.4763308  0.2029545
## p:time2        -2.0214974 0.0464884 -2.1126147 -1.9303800
## p:SexM:time2   0.2462126 0.0686218  0.1117138  0.3807114
##
## 
## 
## Real Parameter p
##           all.diff.index par.index estimate      se      lcl      ucl fixed
## p gF t1            1          1 0.1451876 0.0142288 0.1194587 0.1753545
## p gF t2            2          2 0.0220025 0.0023064 0.0179080 0.0270073
## p gM t1            3          3 0.1290322 0.0146030 0.1030094 0.1604531
## p gM t2            4          4 0.0244868 0.0029329 0.0193509 0.0309429
##
## 
## 
## Real Parameter c
##           all.diff.index par.index estimate      se      lcl      ucl fixed
## c gF t2            5          2 0.0220025 0.0023064 0.0179080 0.0270073
## c gM t2            6          4 0.0244868 0.0029329 0.0193509 0.0309429
nop.sex.time$results$real
##           estimate      se      lcl      ucl fixed note
## p gF t1 0.1451876 0.0142288 0.1194587 0.1753545
## p gF t2 0.0220025 0.0023064 0.0179080 0.0270073
## p gM t1 0.1290322 0.0146030 0.1030094 0.1604531
## p gM t2 0.0244868 0.0029329 0.0193509 0.0309429
nop.sex.time$results$derived

```

```

## $`N Population Size`  

##   estimate      se      lcl      ucl  

## 1 27860.50 2700.203 23140.39 33780.28  

## 2 21524.53 2406.300 17382.88 26878.68

```

The estimated population size is derived from a HT-type estimator for each sex, and we need to find the total, manually:

```

cat("\n\n*** Total population size and se from model sex*time\n")
##  

##  

## *** Total population size and se from model sex*time  

temp <- nop.sex.time$results$derived  

sum(nop.sex.time$results$derived$`N Population Size`[, "estimate", drop=FALSE])
## [1] 49385.03
sqrt(sum(nop.sex.time$results$derived.vcv$`N Population Size`))
## [1] 3616.833

```

20.6 Example - Northern Pike - Model with length as a covariate

We first standardize the length variable and create a new variable for $length^2$.

```

# we standardize the length and 12 variables
data_nop.mark$L <- scale(data_nop.mark$length)
data_nop.mark$L2<- data_nop.mark$L**2
head(data_nop.mark)
##   ch length Sex freq      L      L2
## 1  01  23.20  M    1 -0.7146489 0.5107230
## 2  01  28.89  F    1  0.3710705 0.1376933
## 3  01  25.20  M    1 -0.3330252 0.1109058
## 4  01  22.20  M    1 -0.9054607 0.8198591
## 5  01  25.00  M    1 -0.3711876 0.1377802
## 6  01  24.70  M    1 -0.4284311 0.1835532

```

Now we fit the model:

```

nop.mtLL <- mark( data_nop.mark, invisible=TRUE,
                    model="Huggins",
#                   groups=c("Sex"), # note name must match case and that used in convert.inp
                    model.name="model time*(L+L2)",

```

```

    model.parameters=list(p=list(formula=~time*L+time*L2, share=TRUE))
  )

## 
## Output summary for Huggins model
## Name : model time*(L+L2)
##
## 
## Npar : 6
## -2lnL: 7150.663
## AICc : 7162.668
##
## Beta
##           estimate      se      lcl      ucl
## p:(Intercept) -1.7120266 0.1064005 -1.9205716 -1.5034816
## p:time2        -1.5005710 0.0434446 -1.5857224 -1.4154196
## p:L            0.1734409 0.1311216 -0.0835575  0.4304393
## p:L2           -0.2486398 0.1391710 -0.5214150  0.0241354
## p:time2:L      0.1027038 0.0463942  0.0117712  0.1936364
## p:time2:L2     -0.5414314 0.0453611 -0.6303391 -0.4525237
##
## 
## 
## Real Parameter p
##           1          2
## 0.1233984 0.0179409
##
## 
## 
## Real Parameter c
##           2
## 0.0179409

summary(nop.mtLL, se=TRUE)
## Output summary for Huggins model
## Name : model time*(L+L2)
##
## 
## Npar : 6
## -2lnL: 7150.663
## AICc : 7162.668
##
## Beta
##           estimate      se      lcl      ucl
## p:(Intercept) -1.7120266 0.1064005 -1.9205716 -1.5034816
## p:time2        -1.5005710 0.0434446 -1.5857224 -1.4154196
## p:L            0.1734409 0.1311216 -0.0835575  0.4304393

```

```

## p:L2      -0.2486398 0.1391710 -0.5214150 0.0241354
## p:time2:L 0.1027038 0.0463942 0.0117712 0.1936364
## p:time2:L2 -0.5414314 0.0453611 -0.6303391 -0.4525237
##
##
## Real Parameter p
##      all.diff.index par.index estimate      se      lcl      ucl fixed
## p g1 t1          1          1 0.1233984 0.0124748 0.1009539 0.1500003
## p g1 t2          2          2 0.0179409 0.0019345 0.0145176 0.0221533
##
##
## Real Parameter c
##      all.diff.index par.index estimate      se      lcl      ucl fixed
## c g1 t2          3          2 0.0179409 0.0019345 0.0145176 0.0221533
nop.mtLL$results$real
##      estimate      se      lcl      ucl fixed note
## p g1 t1 0.1233984 0.0124748 0.1009539 0.1500003
## p g1 t2 0.0179409 0.0019345 0.0145176 0.0221533

```

The usual estimates of the population size are obtained:

```

nop.mtLL$results$derived
## $`N Population Size`
##   estimate      se      lcl      ucl
## 1 59207.56 9364.262 43877.66 81052.24

cat("\n\n*** Total population size and se from model\n")
##
##
## *** Total population size and se from model
sum(nop.mtLL$results$derived$`N Population Size`[, "estimate"])
## [1] 59207.56
sqrt(sum(nop.mtLL$results$derived.vcv$`N Population Size`))
## [1] 9364.262

```

We can now obtain predictions of the capture probabilities at the two sampling occasions:

```

pred.length <- seq(from=min(data_nop.mark$L), to=max(data_nop.mark$L), length.out=40)

P.by.length.mtLL <- covariate.predictions(nop.mtLL,
                                             data=data.frame(L=pred.length, L2=pred.length^2), indices=c(1,2))

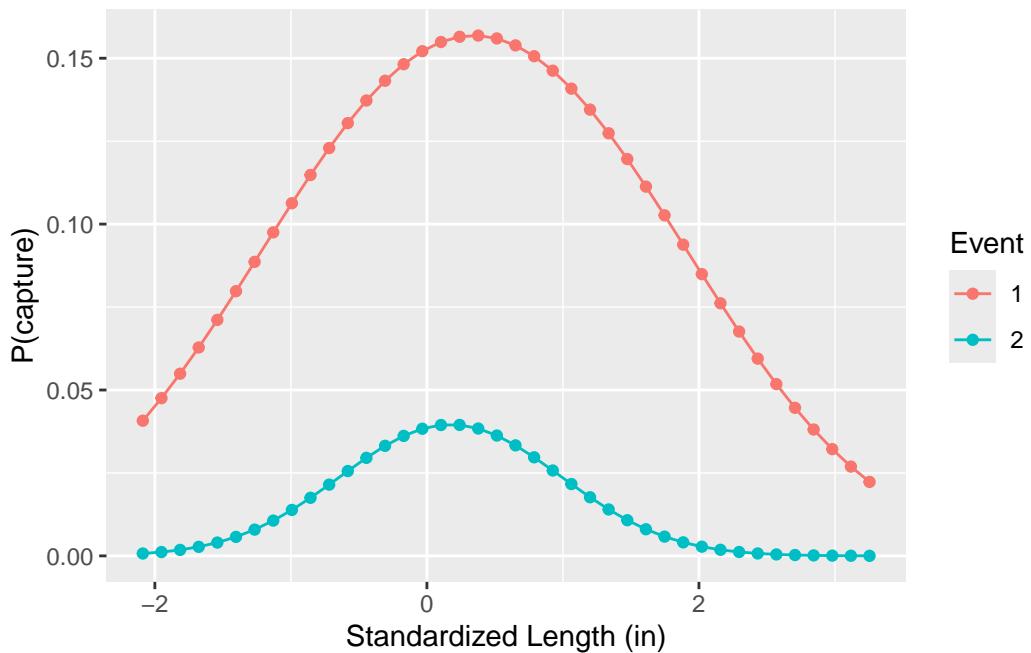
```

```

P.by.length.mtLL$estimates[1:5,]
##   vcv.index model.index par.index      L      L2 estimate      se
## 1          1          1       1 -2.088494 4.361808 0.0407468686 0.0261967645
## 2          2          2       2 -2.088494 4.361808 0.0007200547 0.0004620851
## 3          3          1       1 -1.951501 3.808356 0.0475431991 0.0267132842
## 4          4          2       2 -1.951501 3.808356 0.0011574625 0.0006535587
## 5          5          1       1 -1.814508 3.292439 0.0549212457 0.0267340223
##           lcl         ucl fixed
## 1 0.0112910346 0.136442171
## 2 0.0002046207 0.002530570
## 3 0.0154644068 0.136911559
## 4 0.0003825235 0.003496832
## 5 0.0207373798 0.137540117

ggplot(data=P.by.length.mtLL$estimates, aes(x=L, y=estimate, color=as.factor(par.index)))+
  geom_point()+
  geom_line()+
  xlab("Standardized Length (in)")+ylab("P(capture)")+
  scale_color_discrete(name="Event")

```



21 Appendix D. Conditional likelihood approach to tag-loss studies.

We follow the approach of Hyatt et al (2012) by looking at the case of 2 indistinguishable tags. The cases of distinguishable or a permanent tag is similar and not presented here.

Figure 35 shows the fates of fish in the tag loss model with 2 indistinguishable tags.

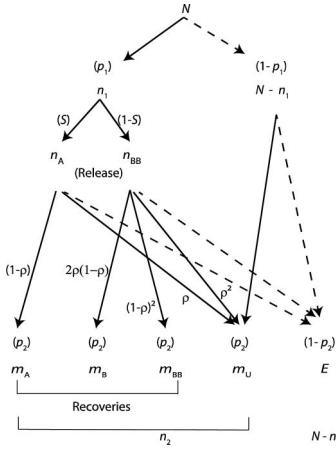


Figure 35: Fates of animals in population under tag-loss model with 2 indistinguishable tags.

In this diagram, the capture probabilities are denoted by p_1 , and p_2 , the probability of a single tag being applied is S , and the tag *loss* probabilities are denoted by ρ . The codes *A* and *B* denote fish with single and double tags applied in the first sample.

The probability of the capture histories are:

- $P(1010) = p_1 S(1 - \rho)p_2 =$ single tag fish recaptured with tag present = m_A
- $P(1000) = p_1 S\rho + p_1 S(1 - \rho)(1 - p_2)$ - single tag fish never seen again (either tag lost or not captured at event 2)
- $P(1111) = p_1(1 - S)(1 - \rho)^2p_2 =$ double tagged fish recaptured with both tags present m_{BB}
- $P(111X) = p_1(1 - S)2\rho(1 - \rho)p_2 =$ double tagged fish recaptured with only tag present. There are two ways in which a single tag can be lost = m_B .
- $P(1100) = p_1(1 - S)\rho^2 + p_1(1 - S)2\rho(1 - \rho)(1 - p_2) + p_1(1 - S)(1 - \rho)^2(1 - p_2) =$ double tagged fish lost both tags or retained at least one tag and was never captured at time 2
- $P(0010) = p_1 S(1 - \rho)p_2 + p_1(1 - S)\rho^2p_2 + (1 - p_1)p_2 =$ probability of an apparently untagged fish being captured at event 2 = m_U .

Notice that some fish can be double counted, i.e. fish that were single tagged and never seen again also includes fish that were captured at event 2 after loosing its tag; and fish that were double tagged and never seen again, also includes fish that were captured at event 2 after

loosing both tags. Consequently, a full multinomial model cannot be constructed because the probabilities add to more than 1.

However, suppose we condition on being captured at event 2. This corresponds to fish with histories 1010, 1111, 111x, and 0010. This has total probability of p_2 .

The conditional probabilities are now

- $P(1010 \mid \text{captured at event 2}) = p_1 S(1 - \rho)$
- $P(1111 \mid \text{captured at event 2}) = p_1(1 - S)(1 - \rho)^2$
- $P(111X \mid \text{captured at event 2}) = p_1(1 - S)2\rho(1 - \rho)$
- $P(0010 \mid \text{captured at event 2}) = p_1 S(1 - \rho)p_2 + p_1(1 - S)\rho^2 p_2 + (1 - p_1)$

Now a conditional maximum likelihood approach can be used to estimate p_1 and ρ in a similar fashion as the conditional maximum likelihood Lincoln-Petersen estimator.

Once estimates of p_1 are obtained, the estimated abundance is formed using a Horvitz-Thompson type estimator

$$\widehat{N} = \sum_{\text{animals captured at event 1}} \frac{1}{\widehat{p}_{1i}}$$

Estimates of the uncertainty of the abundance estimates are formed again similar to those in the conditional likelihood Petersen estimator.

The main advantage of the conditional likelihood approach is that models where the capture/tag retention parameters vary by covariates that are unknown for fish never seen can be fit again similar to the conditional likelihood Petersen estimator.

22 Appendix E: Brief theory for *BTSPAS* models.

The complete theory for the *BTSPAS* models is presented in Bonner and Schwarz (2011). This is a synopsis.

We consider the two cases – the diagonal and non-diagonal cases.

22.1 Diagonal case

In the diagonal model, the relevant data are:

- n_{1i} - the number released in temporal stratum i
- m_{2i} - the number from those releases temporal stratum i that are captured together in a single future temporal stratum

- $u2_i$ - the number of UNmarked fish captured in a future temporal stratum along with the recaptures from releases in temporal stratum i .

The capture history data can be structured as:

We can extract the two temporal strata and make a matrix of releases and recoveries of the form:

Recovery Stratum		Never	rs1	rs2	rs3	...	rsk	Tagged
Newly	seen again							
Untagged			u2[1]	u2[2]	u2[3]	...	u2[k]	
captured								
Marking	ms1	n1[1]-m2[1]	m2[1]	0	0	...	0	n1[1]
Stratum	ms2	n1[1]-m2[2]	0	m2[2]	0	...	0	n1[2]
	ms3	n1[1]-m2[3]	0	0	m2[3]	...	0	n1[3]
	...							
	msk	n1[1]-m2[k]	0	0	0	...	m2[k]	n1[k]

Here the tagging and recapture events have been stratified into k temporal strata. Marked fish from one stratum tend to move at similar rates and so are recaptured together with unmarked fish. Recaptures of marked fish take place along the “diagonal” as shown below for the first 10 temporal strata:

Note that while this model is referred to as the *diagonal model*, it is not strictly necessary that recaptures/captures take place in the same temporal stratum as releases. For example, fish could be released in julian week 21 of a calendar year, but take 2 weeks to move to the next fish wheel for recapture in julian week 23. similarly, fish released in julian week 22, are then recaptured in julian week 24, etc. In this case the two-week offset is “hidden” from the program and we “pretend” that releases and recaptures take place in the same temporal stratum.

The Bayesian model has two components. First, we condition on the number of releases and model the recaptures as Binomial distributions:

$$m_{2i} \sim \text{Binomial}(n_{1i}, p_{2i})$$

This is similar to a fully stratified Petersen estimator. However, we assume that the recapture probabilities come from a hierarchical model on the *logit()* scale:

$$\text{logit}(p_{2i}) \sim \text{Normal}(\mu, \sigma)$$

The advantage of the hierarchical model is that inference for the parameters in one stratum will depend on the data from all strata. For example, if the data suggests that all of the strata have

p 's close to .025, then this information is used to improve estimates for strata with poor data (or no data at all). In this way, data is shared among the strata, but the parameters are still allowed to vary among strata. The disadvantage of the hierarchical approach is that it makes no adjustment for the ordering of the data – the same amount of information is shared between strata 1 and 2 as strata 1 and 10 or strata 1 and 100. Various extensions to the BTSPAS model could be developed to account for this autocorrelation, but in practice, we have not found it necessary. This follows the framework of Mantyniemi and Romakkaniemi (2002).

Second, we model the unmarked fish captured at the “second” event also by a binomial distribution with the same (re)capture probabilities

$$u_{2i} \sim \text{Binomial}(U_{2i}, p_{2i})$$

where U_{2i} is the population abundance of unmarked fish passing the second trap along with fish released in temporal stratum i . Again, this is similar to the fully stratified-Petersen estimator. However, it seems intuitive that the number of fish passing the recapture trap will be more similar for strata that are close together and less similar for strata that are further apart. To account for this structure, we impose smoothness on the U_{2i} using a spline that allows for a very general shape.

Bonner and Schwarz (2011) chose to model U_{2i} explicitly as a smooth curve using the Bayesian penalized spline (P-spline) method of Lang and Brezger (2004). Two factors control the smoothness of a spline: the number and locations of the knots and the variation in the coefficients of the basis-function expansion. The classical P-spline method of Eilers and Marx (1996) approaches this dichotomy by fixing a large number of knot points and then penalizing the first or second order differences of the coefficients. In the original implementation, the spline curve is fit by minimizing a target function which adds the sum-of-squared residuals and a penalty term formed as the product of a smoothing parameter and the sum of the differences of the spline coefficients. Increasing the smoothing parameter places more weight on the penalty term and results in a smoother curve. Decreasing the smoothing parameter places more weight on the sum-of-squared residuals and produces a fit that comes closer to interpolating the data.

Although the spline model may reflect the trend in the daily population size, similar to a running mean, it is unlikely that U_{2i} will exactly follow a smooth curve. If the deviations from a smooth curve are small then it seems reasonable that forcing U_{2i} to be smooth will not have a large impact on the estimation of overall abundance. However, if there are large deviations from the smooth curve then forcing the U_{2j} to be smooth may severely bias the estimate of the total population size. To allow for roughness in the U_{2i} over-and-above the spline fit, the spline model is extended with an additional “error” (extra variation) term to allow the U_{2i} to vary above and below the spline.

Bonner (2008, Section 2.2.4) conducted an extensive simulation study to compare the Bayesian P-spline method with the stratified-Petersen estimator. Generally, the Bayesian P-spline method had negligible bias and its precision was at least as good as the stratified-Petersen

estimator. When “perfect” data were available (i.e., many marked fish were released and recaptured in each stratum) the results from the Bayesian P-spline and the stratified-Petersen were very similar. When few marked fish were released or recaptured in each stratum, the performance of the Bayesian P-spline model depended on the amount of variation between the capture probabilities and the pattern of abundance over time. In the worst case, with large variations between the capture probabilities and abundances that followed no regular patterns, the two models continued to perform similarly. However, when the variation between the capture probabilities were smaller and the abundances followed close to a smooth curve the Bayesian P-spline produced much more precise estimates of the total population size.

22.2 Non-diagonal case

In many cases, the released fish are not recaptured in a single future temporal stratum, and recaptures takes place for a number of future strata.

This gives rise to the matrix of releases and recoveries of the form:

		Recovery Stratum							
Newly		u2[1]	u2[2]	u2[3]	...	u2[k]	u2[k,k+1]		
Untagged		tagged	rs1	rs2	rs3	...rs4	rsk	rs(k+1)	
Marking	ms1	n1[1]	m2[1,1]	m2[1,2]	m2[1,3]	m2[1,4]	0	...	0
Stratum	ms2	n1[2]	0	m2[2,2]	m2[2,3]	m2[2,4]	...	0	0
	ms3	n1[3]	0	0	m2[3,3]	m2[3,4]	...	0	0
	...								
	msk	n1[k]	0	0	0	...	0	m2[k,k]	m2[k,k+1]

In the above representation, released fish take between 0 and 3 weeks to travel between the two traps between release and recapture.

BTSPAS includes a function to fit a log-normal distribution to the length of time that individual fish take to move between the temporal stratum of release and temporal stratum of recovery, but this model may not be sufficiently flexible. Alternatively, *BTSPAS* includes a function that uses a multinomial distribution to describe the movement of fish from the temporal release stratum to the temporal recovery stratum with the restriction that this multinomial movement model slowly changes over time.

GIVE MORE DETAILS HERE

The hierarchical model for the recapture probabilities and the smoothing spline are similar in this non-diagonal non-parametric movement model.