Skeena Sockeye In-river Run Reconstruction Analysis Model

and

Analysis Results for 1982-2009

Prepared for:

Pacific Salmon Foundation

Prepared by:

Karl K. English, Cameron Noble, Anita C. Blakley

with substantial assistance from

William J. Gazey and Steve Cox-Rogers

LGL Limited
environmental research associates
9768 Second Street
Sidney, BC, V8L 3Y8

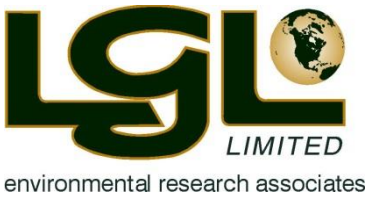

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF APPENDICES

INTRODUCTION

The Fraser, Skeena and Nass watersheds are the three largest sockeye producing watersheds in British Columbia. Exploitation rate estimates for the Nass and Skeena Sockeye stock aggregates are estimated annually using the Northern Boundary Sockeye Run Reconstruction (NBSRR) Model (English et al. 2004; 2005; Alexander et al. 2010). English et al. (2012) provided estimates of marine exploitation rates for each Nass and Skeena sockeye Conservation Unit (CU) using estimates of the migration timing for each CU. These analyses have not included the details on the location and timing of in-river fisheries needed to estimate harvests for the various sockeye CUs or sub-stocks within each watershed. In some years, in-river harvest account for a large portion of the Canadian harvest of sockeye returning to these rivers. Run reconstruction analyses have been used to estimate CU-specific harvest rates for in-river fisheries targeting Fraser Chinook and sockeye salmon (English et al. 2007; Noble 2011). This report provides a brief outline of a Skeena Sockeye In-River (SSIR) run reconstruction model built to combine information on run timing and escapements for Skeena sockeye sub-stocks with catch estimates for each sockeye fishery within the Skeena watershed. The model is similar to those developed for the Fraser River sockeye fisheries within the Fraser watershed except that the Skeena model moves fish forward (upstream) through the fisheries and subtracts sockeye catches from estimates of the number of sockeye entering the Skeena River each day. The Fraser Chinook and sockeye models reconstruct the runs entering the Fraser River by adding catches to daily estimates of escapement for each sub-stock (English et al. 2007; Noble 2011).

Analysis Objectives

Estimates of in-river harvest by sub-stock are needed to be combined with those for marine fisheries to estimate total exploitation rates for Skeena sockeye. The SSIR model provides a systematic process for combining information on catch, fishery timing, stock-specific migration rates through fisheries escapement and river entry run timing by sub-stock. The results from these run reconstruction analyses will be combined with the marine harvest rates from NBSRR model to provide estimates of the total exploitation rate for each Skeena sockeye sub-stock and the biological basis for the forward looking model needed to evaluate alternative fisheries and fisheries management options for Skeena sockeye sub-stocks.

DATA SOURCES AND PREPARATION

Sockeye Stocks and Stock Aggregates

The model has the capacity to accommodate details for as many sockeye sub-stocks or run-timing groups as can be defined using the available data and combine these stocks into any number of management groups. Initial analyses of available run-timing data resulted in the definition of 20 sub-stocks for the Skeena watershed (English et al. 2012). Each of these sub-stock groups contains one or more of the Conservation Units (CUs) defined in Holtby and Ciruna (2007). Where the run-timing and geographic location of the spawning area is not different for two or more CUs, these CUs were combined into a single sub-stock group. The "+" at the end of a sub-stock name is used to indicate that more than one CU is associated with that sub-stock group (Table 1). The only exception in the Zymoetz sub-stock which does not have a "+" but includes three CUs (Mcdonell, Aldrich, Dennis). It should also be noted that the Babine Lake CU has been sub-divided into 5 sub-stock groups based on run-timing and level of enhancement. The two enhanced streams with major spawning channel (Pinkut and Fulton) have slightly different run-timing than the three wild Babine run-timing groups (Table 1).

Fishery Definitions

The SSIR model includes two types of fisheries (FSC and ESSR) and 12 fishing areas: 3 Tsimshian fishery strata in the lower Skeena and 9 fishing areas used by other Skeena First Nations above Fiddler Creek (Table 1). All of the fisheries on the Skeena River (mainstem) harvest multiple stocks and stock composition estimates are not available for these fisheries. Consequently, run reconstruction analysis is required to distribute the reported weekly catches between the stocks vulnerable to each fishery. Some FSC and most ESSR fisheries occur in locations where only a single stock is affected. The harvest rate estimates for these fisheries were computed by dividing the catch by the sum of the annual catch and escapement for these stocks.

Escapement and Entering Run Size

Annual escapement estimates for each Skeena sockeye CU were combined with daily Tyee test fishery data and stock-specific run timing parameters to produce estimates of daily escapement past the Tyee test fishery for each CU. Annual estimates of spawning escapement for the 20 Skeena CU-run timing groups were derived from three sources: 1) nuSEDS data for 11 non-Babine sockeye stock groups; 2) DFO Prince Rupert historical databases for the 5 Babine sockeye stock groups (Cox-Rogers, DFO, pers. comm.); and 3) assumed fixed values for the remaining 4 non-Babine sockeye stock groups without escapement monitoring programs. Each of the non-Babine sockeye stocks had one or more years of missing escapement estimates and these were filled in by using the average of the estimates for adjacent years or interpolating between the available estimates. The filled in values are highlighted in yellow in Table 2. Escapement estimates for the enhanced Pinkut and Fulton sub-stocks included escapements surplus to the spawning channel capacities. The annual escapement estimates for sub-stocks with terminal fisheries (i.e. Pinkut, Fulton, Sustut and Bulkley-Morice) were increased to account for catches in these terminal fisheries prior to determining the portions that each sub-stock proportions represents of the total return of Skeena sockeye in a given year. These sub-stock proportions were combined with annual estimates of the total sockeye abundance passing Tyee and run-timing parameters derived from analysis of 2000-10 Tyee DNA samples (Cox-Rogers, DFO Rupert, pers. comm.) to compute the daily abundance passing Tyee for each sub-stock.

Run Timing

Estimates of river entry timing for 20 sub-stocks of Skeena sockeye were obtained from information reported in a memorandum entitled "SKEENA SOCKEYE SUB-STOCK RUN-TIMING AND ABUNDANCE EVALUATED USING TYEE TEST FISHERY DNA: 2000-2010" prepared by Steve Cox-Rogers dated 23 February 2012. The relative timing for each sub-stock was used to determine the offset difference between the average annual timing for all Skeena sockeye and that for a specific sub-stock. For example: Lakelse sockeye were estimated to have a timing 3 weeks earlier than the median timing for Skeena sockeye, therefore, the offset parameter was set at -21 days for Lakelse sockeye. The duration of the run for each sub-stock was also derived from the 2000-2010 DNA stock composition estimates. The parameter used to define the duration was the standard deviation (SD) measured in number of days. Table 3 provides the offset and SD parameters for each sub-stock. The run timing curve for each sub-stock was defined by a normal curve where the mid-point was defined by combining the stock-specific offset with the median date of sockeye migration past the Tyee test fishery and the start and end points for each timing curve were 3 SD units each side of the mid-point. Therefore, the SD of 13.3 d for Lakelse sockeye results in a total duration of 80 days for this stock. For our initial analysis we used the same offset and SD parameters for each year except 2006, when the duration of the run (SD) for the two large enhanced stocks (Pinkut and Fulton) was increased from 14 to 17.5 days (total duration increased from 84 to 105 days) in order to reflect the notably longer duration of the Skeena sockeye run observed in 2006.

Fishery Residence Time

Residency time was defined as the number of days (to the nearest day) a stock resides within the boundaries of a single fishery. These residence times were derived from historical tagging studies, the differences between peak abundances estimated at Tyee and the Babine fence, and information on the size (river kms) and location of each fishery (English et al. 1985, Steve Cox-Rogers, pers. comm.).

Catch

Estimates of annual harvest by Skeena River First Nations for each fishery location and type from 1982-2009 were initially obtained from DFO records (Table 4, Steve Cox-Rogers, pers. comm.). Estimates of the annual harvests were available for most FSC fisheries from 1996-2009. FSC catch estimates were not available for two FSC fisheries (Kitsegass and Sustut) prior to 1993 and catch estimates for Tsimshian fisheries in the Coastal to Kasiks stratum were substantial underestimates from 2003-09. There were also notable missing FSC estimates for a large portion of the Tsimshian fisheries in 1992 and 2001, Lake Babine FN fisheries in 1987 and the Bulkley River (Wet'suwet'en) fisheries in 1984 and 1990. There were concerns regarding the accuracy of the FSC estimates for the Kitsumkalum fishing area (Kasik-Terrace), the distribution of the GWWA catch between the mainstem Skeena fisheries above and below Hazelton, and FSC catch estimates for fisheries above the Babine fence. All of these issues, except the missing catch estimates for Kitsegass and Sustut fisheries, were addressed using the best available data from a variety of sources. The shaded cells in Table 5 identify all the FSC catch estimates that were adjusted. The footnotes for Table 5 (expanded below) describe how these adjustments were made:

1. Coastal to Kasiks catches for 2001 and 2003-09 were estimated by multiplying the adjusted total FSC catch by the average portion (24.7%) that Coastal to Kasiks catches represented of the reported catch for 1996-09.
2. Kasiks to Terrace (Kalum fishery) catches were expanded by 1.96 for all years except 1993 and 2009. This expansion factor was derived by comparing the Kalum catch numbers derived from Kalum fishing permits (the usual catch reporting method) with those derived from the more rigorous catch monitoring efforts conducted in 2007 and 2008 as part of a Treaty Related Measures (TRM) project.
3. Catch estimates for Fiddler to Hazelton and Hazelton to Lower Babine for 2000-09 were derived from tables provided by the Skeena Fisheries Commission (SFC). The average portion that the Fiddler to Hazelton catch represented of the total catch for these two strata (48%) was applied to the annual totals for these fisheries to derive the catch estimates for these two fisheries for each year from 1982-1999.
4. Previous catch estimates for the 1984-92 FSC fisheries above the Babine fence were replaced with values derived from Babine stock assessment tables provided by Steve Cox-Rogers.
5. Missing catch estimates for the Bulkley-Morice fishery in 1984 and 1990 were estimated using the average of the catch in adjacent years.
6. The total SFC catch for 2001 and 2003-09 needed to calculate the catch for the Coastal to Kaskis stratum was estimated by expanding the total reported catch for fisheries above Kaskis by 1.328 (i.e. 1/0.753) because, on average, fisheries above Kasiks represented 75.3% of the total FSC harvest of Skeena sockeye.

Catch estimates from ESSR commercial fisheries were first conducted on the Skeena River in 1993. These fisheries are only permitted at specific locations within the Skeena watershed when managers determine that Escapements Surplus to Spawning Requirements (ESSR) can be harvested for some Skeena sockeye stocks. In recent years, additional commercial fishing opportunities have been provided to Skeena River First Nations through the transfer of sockeye allocations for Area 4 seine and gillnet licences to in-land "demonstration" fisheries. From 1993-2009, there have been 10 years when ESSR

and/or demonstration fisheries have been conducted within the Skeena watershed and total harvests in these fisheries have ranged from 13,700 to over 780,000 sockeye (Table 6).

The breakdown of these annual harvest estimates to weekly harvests was not available for most fisheries, so the available annual harvest estimates were prorated to weekly harvest estimates using the year-specific estimates of the number of sockeye passing Tyee each week adjusted for the time required for sockeye to migrate from Tyee to each fishery. Ideally, these initial estimates of weekly catch will be replaced with the best available estimates from Skeena First Nations. If weekly estimates are not available of some in-river fisheries, additional information on the timing and duration of these fisheries could be used to improve the estimate of weekly catches. Adjustments to the timing of FSC harvests are not expected to have a significant impact on the harvest rate estimates because of the small relative magnitude and protracted nature of these fisheries. The timing of the more substantial ESSR fisheries along the Skeena mainstem from Fiddler Creek to the Babine fence could have a significant impact on harvest rates for sockeye stocks migrating through these fisheries.

Fishing Patterns

Detailed information on fishing patterns (number of fishing days per week) for Skeena First Nation fisheries have not been obtained so the SSIR model currently runs on the assumption that weekly First Nation catches are distributed equally across all days in a week.

RUN RECONSTRUCTION MODEL

The theoretical basis of run reconstruction analysis for salmon stocks and fisheries are described in Starr and Hilborn (1988), Cave and Gazey (1994), Gazey and English (2000) and English et al. (2007). The SSIR model uses similar algorithms as those described in the 2007 PSARC approved run reconstruction model for Fraser Chinook (English et al. 2007). The sequential steps in the run reconstruction are described below:

1. read all catch, escapement, run-timing parameters and total daily abundance estimates derived from Tyee test fishery data;
2. estimate the daily escapement past the Tyee test fishery for each sub-stock;
3. starting with the first fishery in the lower Skeena, assign portions of the weekly catch to each sub-stock present in the fishery using the estimated constant daily harvest rate for all days in a week based on the assumption of equal vulnerability for all sub-stocks present during the week;
4. subtract the catch for each stock from the abundance of that stock that entered the fishery;
5. repeat steps 3 and 4 for each fishery moving upstream along the Skeena mainstem and into the tributaries;
6. total the catch and escapement estimates for each sub-stock and calculate annual estimates for the in-river harvest rates for each sockeye sub-stock and management group.

The model control worksheet has locations where the user can define the start and end years for the analysis and input files to be used for the run reconstruction analyses.

A more mathematically rigorous description of the above methods can be found in English et al. (2007).

Model Assumptions

The assumptions associated with the SSIR run reconstruction analyses model include:

a. The sockeye sub-stocks included in the models adequately represent the run timing and total escapement for Skeena sockeye;
b. The daily sockeye CPUE estimates from Tyee test fishery provides a reliable indication of the relative abundance sockeye entering the Skeena River;
c. The escapement estimates and run-timing parameters available for Skeena sub-stocks can be used with the assumption of normal distributions for each stock to derive daily stock composition estimates for the run at Tyee;
d. The fisheries and catch data included in the model adequately represent the timing and location of fisheries that harvest sockeye within the Skeena watershed; and
e. All stocks are equally vulnerable to harvesting when present in a fishery, such that harvests of a stock are proportional to the relative abundance of that stock in that fishery during the fishing period.

Model Structure

In order to expedite these run reconstruction analyses, we have used a model structure that is very similar to that used for the Fraser Chinook run reconstruction model (i.e. a MS Excel model prepared using the Visual Basic programming language). The model contains a series of sub-routines and function calls to read input data from MS Excel worksheets, conduct the analyses and output results to MS Excel files.

The model includes the following sub-routines and functions:

Sub Reconstruction() - main program where all other sub-routines are called.

Sub Init() – reads the year range and input file names from the "Control" worksheet, opens the input files, creates the output files and writes the initial column headings into each output file.

Sub Read_Catch() – reads the weekly catch data for each FSC and ESSR fishery for the range of years included in the analysis.

Sub FishResSpawn() – reads the fishery residence times for each stock and determines the cumulative number of days between each fishery and the escapement area for each stock.

Sub Calc_Escape() – calculates the daily escapement for each stock for a specific year.

Sub Reconstruct() - conducts the run reconstruction analysis working backward through the fisheries building on the daily escapement estimates.

Function CalcHarvestRate() - calculates the weekly harvest rate for a given fishery based on the size of the reported catch, number of fishing days per week and the number of sockeye that escaped from that fishery.

Function gfs() – calculates the weekly catch for a given harvest rate. This function is used in the bisection algorithm to determine the weekly harvest rate that would result in the reported catch.

Sub OutputData() – writes the run reconstruction results to the various output files defined below.

Model output file name and description:

Catch.xls	• Annual total catch by stock group (Babine and non-Babine), annual FSC catch by stock group and by stock, annual ESSR catch by stock group and stock.
FisheryHR.xls	• Daily HR by fishery, week and year (separate worksheet for each year) computed to derive the weekly catch estimates for each stock.
WeeklyHR.xls	• Mean estimated daily HR by fishery, week and year. These estimates are identical to those in FisheryHR.xls when fisheries are open 7 days a week.
ExploitRate.xls	• Annual exploitation rates by stock group and by stock for each year
Reconstruction.xls	• Annual summary of catch, for each stock and fishery (worksheet for each year).
Escape.xls	• Annual escapement at spawning grounds by stock group and stock.
Unexplained.xls	• Unexplained catch (not reconstructed) by fishery, week and year.

The internally documented source code for the current version of the SSIR model is provided in Appendix B.

RESULTS

Figures 1-3 provide a sample of the run-timing and abundance of sockeye passing the Tyee test fishery in the lower Skeena River for 2006-08. These figures also show the normally distributed run-timing curves for each of the major sub-stock groups and the resulting breakdown of the total Tyee abundance for each of the major sub-stocks. These years were selected because 2006 is an example of one of the latest run-timing years, 2007 run-timing is close to the multi-year mean and 2008 is one of the earliest run-timing years. As indicated above, 2006 was the only year where we increased the duration of two sub-stocks (Pinkut and Fulton) to reflect the protracted nature of the sockeye return in that year and ensure that the stock composition estimates for Tyee were consistent with the best available escapement estimates.

These figures clearly show the substantial overlap in the run-timing and long durations estimated for most Skeena sockeye stocks. These long durations are likely the result of having to use multiple years of DNA samples to obtain an adequate sample size for the relatively small non-Babine sockeye stocks. Steve Cox-Rogers 23 February 2012 memorandum included the following conclusion:

> *"The estimated peak dates of run entry for most Skeena sockeye sub-stocks, based on updated 2000-2010 DNA analysis, are not substantially different from past tagging assessments and the peak dates currently being used to assess stock impacts. The DNA data does suggest slightly wider "spreads" about the peaks for most stocks than currently assumed, and some apparent skewness/bi-modal variability to the timings may not be appropriately captured with the current practice of fitting normal curve approximations to the data. However, it is not clear how much of the shape variation is real or simply an artefact of sample size issues given the small number of DNA samples actually analyzed for some stocks in certain weeks (e.g. the tails of the test fishery). This, coupled with the fact that many non-Babine stocks are present in small proportions at Tyee in the first place, means the derived timings for the larger stocks are probably ok, but will always be uncertain for the smaller ones."*

While it is likely that the run duration for a single year would be shorter than the duration derived from samples collected over multiple years, the harvest rate estimates derived from these longer run durations will be less sensitive to uncertainties in the run and harvest timing for a given year. In the absence of more reliable year specific data on run-timing and duration, a conservative approach for estimating harvest rates for in-river fisheries is to use these longer durations.

Two model inputs that are critical for deriving reliable estimates of in-river harvest rates are catch estimates for all major fisheries and relative escapement estimates for each sub-stock. Our initial analyses revealed several deficiencies in the available catch and escapement estimates that have been specifically addressed for this report. Additional information on sockeye harvests from Tsimshian fisheries managers, SFC and DFO has been used to fill gaps and errors in the time-series of catch estimates for each of the major FSC sockeye fisheries (Table 5). The likely underestimation of escapements to the Bulkley-Morice watershed prior to 1989 has been flagged as an issue that still needs to be addressed. These escapement estimates have resulted in substantial overestimates of the harvest rates for the Moricetown fishery for 1982-1988 (Table 7). For all other sockeye stocks, the SSIR model has produced time-series of in-river harvest rates that are consistent with all the available data on the magnitude and location of fisheries and run-timing and geographic distribution of the sockeye sub-stocks. For example: in 2000 when ESSR fisheries were permitted to target surplus escapements for enhanced Babine stocks, in-river harvest rates were 43-45% for Pinkut and Fulton, 20-31% for the three run-timing groups of wild Babine sockeye, 23-27% for upper Skeena sockeye stocks and 1-5% for the early run lower Skeena sockeye stocks (Table 7). In years without ESSR fisheries (e.g. 2002 and 2005), in-river harvest rates are similar (16-23%) for all Babine sockeye sub-stocks, 14-17% for upper Skeena stocks and 2-7% for lower Skeena stocks. The two lower Skeena stocks with notably shorter durations and least overlap with the Babine enhanced stocks are Lakelse and Zymoetz. The in-river harvest rates for these stocks were estimated to be 2-5% for most years.

DISCUSSION

Increasing levels of harvest in fisheries within the Skeena watershed have made these fisheries a significant component of the annual exploitation rates for many Skeena sockeye CUs. The location of stocks and fisheries within the watershed can result in harvest rates that differ substantially between the various sockeye CUs. In general, the harvest rates tend to be larger for the Babine sub-stocks because of the additional fisheries within the Babine watershed that target the surplus returns to the enhanced stocks. However, the highest harvest rates estimated for Skeena sockeye CUs were for the Bulkley/Morice CUs prior to 1989 and the Sustut CU after 2004. The bulk of the in-river harvest for these CUs occurred in terminal fisheries within the Bulkley or Sustut watersheds. The unusually high harvest rates estimated for sockeye in the 1982-88 Bulkley River fisheries has been cause for concern that the catch estimates, escapement estimates or both may be unreliable for this period. Sustut was the only other sockeye CU with unusually high harvest rates. The Sustut escapement estimates were derived from fence counts and are believed to be very reliable. The catch numbers are also believed to be reliable. Given the location of the fishery and relatively small returns in most since 2005, harvest rates estimates in the 50-75% range for in-river fisheries are not unrealistic. The Babine enhanced stocks are the only other Skeena sockeye stocks with in-river harvest rates above 35% and these high harvest rates only occur in years when major ESSR fisheries are conducted in Babine Lake near the Pinkut and Fulton spawning channels.

LITERATURE CITED

Alexander, R., K.K. English, D. Peacock, and G. Oliver. 2010. Assessment of the Canadian and Alaskan Sockeye Stocks harvested in the northern boundary fisheries using run reconstruction techniques, 2004-08. Draft report for Pacific Salmon Comm. Northern Boundary Technical Committee.

Cave J. and W.J. Gazey. 1994. A simulation model for fisheries on Fraser River sockeye salmon. J. Fish. and Aquat. Sci. 51:1535-1549.

Cox-Rogers, S. 2012. Skeena sockeye sub-stock run-timing and abundance evaluated using Tyee test fishery DNA: 2000-2010. Fisheries and Oceans Canada Memorandum dated 23 February 2012. 23 p.

English, K.K., T. Mochizuki and D, Robichaud. 2012. Review of North and Central Coast Salmon Indicator Streams and Estimating Escapement, Catch and Run Size for each Salmon Conservation Unit. Report for Pacific Salmon Foundation and Fisheries and Oceans, Canada. 68 p.

English, K.K., R. E. Bailey, and D. Robichaud. 2007. Assessment of Chinook returns to the Fraser River watershed using run reconstruction techniques, 1982-04. Canadian Science Advisory Secretariat, Research Document 2007/020. 76 p.

English, K.K., R. Alexander, D. Peacock, and G. Oliver. 2005. Assessment of the Canadian and Alaskan Sockeye Stocks harvested in the northern boundary fisheries using run reconstruction techniques, 2002-03. Prepared for Pacific Salmon Comm. Northern Boundary Technical Committee. 59 p.

English, K.K., W. J. Gazey, D. Peacock, and G. Oliver. 2004. Assessment of the Canadian and Alaskan Sockeye Stocks harvested in the northern boundary fisheries using run reconstruction techniques, 1982-2001. Pacific Salmon Comm. Tech. Rep. No. 13:93 p.

English, K.K., D. Hall, and J.A. Taylor. 1985. The North Coast Salmon Tagging Project. Management information introductory volume, guide to figures and tables. 38 p.
Volume A, 1982 Sockeye Salmon, 105 p.

Volume B, 1982 Pink Salmon, 111 p.

Volume C, 1983 Sockeye Salmon, 129 p.

Volume D, 1984 Pink Salmon, 131 p.

Unpublished report by LGL Limited for Fisheries and Oceans, Canada.

Gazey, W.J. 2009. Interception of Skeena River Sockeye salmon stocks in northern boundary marine fisheries. Report for Skeena Wild Conservation Trust, Terrace, BC. 43 p.

Gazey, W.J., and K.K. English. 2000. Assessment of sockeye and pink salmon stocks in the northern boundary area using run reconstruction techniques, 1982-95. Can. Tech. Report Fish. Aquat. Sci. No. 2320. 132 p.

Noble, C. 2011. Assessing the performance of an in-river backward run reconstruction of Fraser River sockeye under biological uncertainty. MRM Thesis. Simon Fraser University. 53 p.

Starr, P. and R. Hilborn. 1988. Reconstruction of harvest rates and stock contribution in gauntlet salmon fisheries: application to British Columbia and Washington sockeye (*Oncorhynchus nerka*). Can J. Fish. Aquat. Sci 45: 2216-2229.

Table 1. Fisheries, stocks and estimated residence time in days for the Skeena sockeye in-river run reconstruction analyses.

						Fisheries											
Stocks (Geographic CUs)	Short Name	Order	Aggregate	Data Quality[5]	CUs in Group	Coastal to Kasiks	Kasiks-Terrace	Terrace-Fiddler	Fiddler-Hazelton	Hazelton-L Babine	Babine below Fence	Babine Fence	Babine Lake	Pinkut Terminal	Fulton Terminal	Bulkley-Morice	Sustut
Kluatantan/Kluayaz	Kluatan+	1	1	W	2	3	4	3	4	7							
Motase	Motase	2	1	W	1	3	4	3	4	7							
Sustut/Johanson/Spawning	Sustut+	3	1	G	3	3	4	3	4	7							1
Bear/Azuklotz/Asitka	Bear+	4	1	G	3	3	4	3	4	7							
Slamgeesh/Damshilgwit	Slamgeesh	5	1	G	2	3	4	3	4	7							
Sicintine	Sicintine	6	1	W	1	3	4	3	4	7							
Babine W Early[1]	Babine-WE	7	2	G	1	3	4	3	4	7	4	1	3				
Babine W Middle[2]	Babine-WM	8	2	G	1	3	4	3	4	7	4	1	3				
Babine W Late[3]	Babine-WL	9	2	G	1	3	4	3	4	7	4	1	3				
Babine Pinkut	Babine-P	10	2	G	1	3	4	3	4	7	4	1	3	1			
Babine Fulton	Babine-F	11	2	G	1	3	4	3	4	7	4	1	3		1		
Swan/Stephans/Club	Swan+	12	1	G	3	3	4	3	4								
Bulkley/Maxan	Bulkley+	13	1	W	2	3	4	3	4							1	
Morice/Atna	Morice+	14	1	G	2	3	4	3	4							1	
Kitwanga	Kitwanga	15	1	G	1	3	4	3	2								
Zymoetz[4]	Zymoetz	16	1	W	3	3	4	1									
Kalum	Kalum	17	1	W	1	3	4										
Lakelse	Lakelse	18	1	G	1	3	4										
Alastair	Alastair	19	1	G	1	3											
Johnston/Ecstall	Johnston	20	1	W	1	1											

[1] Babine W Early includes sockeye spawing in non-enhanced tributaries to Babine Lake and in Onerka Lake.

[2] Babine W Middle includes the Tahlo/Morrison CU.

[3] Babine W Late includes the Nilkitkwa Lake CU.

[4] Zymoetz includes three sockeye lake CUs in the Zymoetz watershed (Mcdonell, Aldrich and Dennis).

[5] G= Good, W=Weak

Table 2. Model input spawning escapement estimates for each sub-stock of Skeena sockeye salmon 1982-2010, where the estimates for Babine sub-stocks are prior to removals in fisheries at or above the Babine fence.

	Kluatan+	Motase	Sustut+	Bear+	Slamgeesh	Sicintine	Babine-WE	Babine-WM	Babine-WL	Babine-P	Babine-F	Swan+	Bulkley+	Morice+	Kitwanga	Zymoetz	Kalum	Lakelse	Alastair	Johnston	Skeena Agg.
1982	1,000	500	4,000	947	1,000	1,000	93,630	5,195	159,595	242,851	635,565	8,443	1,000	6,375	1,880	2,000	758	30,296	9,500	504	1,206,037
1983	1,000	500	4,000	922	1,000	1,000	26,965	9,226	103,027	197,094	550,081	9,498	1,000	8,500	2,720	10,000	606	19,365	13,000	705	960,210
1984	1,000	500	4,000	897	1,000	1,000	26,503	8,335	204,447	366,909	446,191	10,553	1,000	6,375	3,560	1,000	785	9,573	8,000	705	1,102,333
1985	1,000	500	4,000	2,709	1,000	1,000	75,649	17,696	623,637	598,005	833,056	9,287	1,000	4,250	4,400	1,200	1,570	36,530	8,000	907	2,225,396
1986	1,000	500	4,000	2,992	1,000	1,000	26,865	4,115	167,437	224,106	278,984	10,553	1,000	6,375	3,720	6,000	1,570	9,022	21,000	353	771,592
1987	1,000	1,500	4,000	10,917	1,000	1,000	38,206	16,344	237,400	678,196	337,706	21,107	1,000	8,500	3,040	6,000	3,139	5,336	10,000	604	1,385,995
1988	1,000	100	4,000	7,221	1,000	1,000	42,435	24,382	241,974	384,898	715,191	25,328	1,000	2,125	2,360	4,000	1,884	10,508	13,000	705	1,484,109
1989	1,000	400	4,000	1,667	1,000	1,000	18,412	8,005	132,563	239,008	734,328	8,443	1,000	11,900	1,680	3,500	2,825	7,724	14,932	302	1,193,690
1990	1,000	60	4,000	2,279	1,000	1,000	21,328	7,677	198,864	208,061	542,716	10,553	1,000	12,750	1,000	3,000	4,133	2,707	10,000	458	1,033,586
1991	1,000	300	4,000	7,551	1,000	1,000	58,719	26,200	432,582	450,512	208,305	14,775	1,000	85,001	1,000	1,200	4,133	13,534	22,000	613	1,334,424
1992	1,000	500	4,000	5,097	1,000	1,000	52,358	9,455	582,914	226,003	363,054	21,107	1,000	57,376	1,000	10,000	11,022	10,453	16,000	1,992	1,376,332
1993	1,000	400	4,000	5,599	1,000	1,000	16,646	28,016	595,377	539,525	557,861	16,885	1,000	46,751	1,000	15,000	9,645	13,754	15,000	3,371	1,872,830
1994	1,000	250	4,000	8,057	1,000	1,000	25,124	8,070	132,299	214,508	672,904	12,453	1,000	60,563	1,000	10,295	15,156	3,136	13,000	4,750	1,189,565
1995	1,000	250	4,000	10,516	1,000	1,000	79,679	7,351	69,506	603,037	977,436	8,020	1,000	74,376	500	10,295	13,434	27,618	17,000	6,129	1,913,146
1996	1,000	100	4,000	6,308	1,000	1,000	60,909	11,800	143,305	644,740	1,139,837	7,598	1,000	87,126	250	10,295	5,236	23,822	25,000	5,108	2,179,432
1997	1,000	220	4,000	3,816	1,000	1,000	92,245	43,995	129,975	224,874	595,521	5,910	1,000	51,001	250	10,295	10,058	3,466	24,000	4,086	1,207,711
1998	1,000	500	4,000	1,578	1,000	1,000	43,130	17,150	97,880	100,030	252,055	5,910	1,000	12,750	250	10,295	10,747	5,409	11,000	6,129	582,814
1999	1,000	500	4,000	9,195	1,000	1,000	63,692	24,171	155,040	141,127	222,106	4,221	1,000	31,875	250	10,295	12,318	7,235	2,000	8,172	700,197
2000	1,000	400	4,000	5,297	949	1,000	84,558	30,056	185,020	278,370	1,253,611	5,823	1,000	6,375	231	10,295	13,888	7,235	6,200	1,500	1,896,806
2001	1,000	200	4,000	4,348	855	1,000	232,802	85,174	617,401	208,119	840,765	7,425	1,000	8,500	221	10,295	8,292	9,061	10,800	4,500	2,055,757
2002	1,000	100	4,000	897	398	1,000	29,324	26,094	130,726	101,045	308,038	2,533	1,000	14,875	978	7,072	11,072	7,468	4,000	2,000	653,620
2003	1,000	2,000	4,992	11,253	430	1,000	55,028	86,328	99,284	224,981	704,737	5,070	1,000	21,250	3,377	9,106	28,383	7,468	27,000	5,050	1,298,738
2004	1,000	600	1,604	2,998	293	1,000	39,546	54,315	249,231	107,514	468,643	5,538	1,000	16,469	1,317	6,332	13,968	5,875	20,074	2,395	999,714
2005	1,000	290	1,175	2,499	216	1,000	25,141	33,241	163,178	137,759	349,880	2,550	1,000	17,002	937	6,888	8,939	6,305	13,147	2,395	774,542
2006	1,000	120	808	2,849	331	1,000	40,874	16,502	137,660	328,315	868,328	4,529	1,000	17,536	5,139	6,116	9,521	3,632	4,800	2,395	1,452,455
2007	1,000	300	2,469	3,199	366	1,000	52,862	13,376	100,762	239,849	643,632	2,090	1,000	28,475	245	3,800	12,455	6,624	22,000	2,395	1,137,899
2008	1,000	100	212	8,577	150	1,000	28,667	17,536	93,158	264,543	679,415	4,475	1,000	19,125	1,200	280	15,349	5,513	1,119	2,395	1,144,813
2009	1,000	410	540	6,787	161	1,000	20,503	20,163	93,791	157,109	380,436	3,466	1,000	24,342	3,047	3,400	19,521	7,574	21,500	2,395	768,145
2010	1,000	592	426	12,210	740	1,000	20,455	6,520	74,126	145,310	392,644	266	1,000	7,831	20,804	2,980	30,496	14,720	33,700	2,395	769,214
After Fill																					
Average	1,000	442	3,362	4,908	1,000	1,000	50,658	22,134	214,920	291,396	578,934	8,581	1,000	24,475	2,213	6,275	9,097	11,827	14,470	2,465	1,249,960
Portion	0.001	0.000	0.003	0.004	0.001	0.001	0.041	0.018	0.172	0.233	0.463	0.007	0.001	0.020	0.002	0.005	0.007	0.009	0.012	0.002	
Before Fill																					
Average	0	441	1,528	5,080	444	0	50,658	22,134	214,920	291,396	578,934	8,709	0	23,796	3,078	4,939	8,990	12,490	14,283	2,395	1,244,215
Portion	0.001	0.000	0.003	0.004	0.001	0.001	0.041	0.018	0.173	0.234	0.465	0.007	0.001	0.019	0.002	0.004	0.007	0.010	0.011	0.002	

Table 3. Run-timing parameters used for each Skeena sockeye sub-stock group.

#	Stocks (Geographic CUs)	CUs in Group	Offset (days)	Duration (days)	Default SD (days)	2006 SD (days)	Source: Cox-Rogers (2012) Group Name	Peak Week	Offset (days)	SD (weeks)
1	Kluatantan/Kluayaz	2	-10.5	105	17.5	17.5	Bulkley-Morice	72	-10.5	2.5
2	Motase	1	3.5	92	15.4	15.4	Motase	74	3.5	2.2
3	Sustut/Johanson/Spawning	3	-3.5	84	14.0	14.0	Sustut	73	-3.5	2.0
4	Bear/Azuklotz/Asitka	3	-3.5	84	14.0	14.0	Sustut	73	-3.5	2.0
5	Slamgeesh/Damshilgwit	2	-3.5	84	14.0	14.0	Sustut	73	-3.5	2.0
6	Sicintine	1	-3.5	84	14.0	14.0	Sustut	73	-3.5	2.0
7	Babine W Early[1]	1	-10.5	84	14.0	14.0	Babine WE	72	-10.5	2.0
8	Babine W Middle[2]	1	-3.5	84	14.0	14.0	Babine WM	73	-3.5	2.0
9	Babine W Late[3]	1	10.5	84	14.0	14.0	Babine WL	75	10.5	2.0
10	Babine Pinkut	1	-3.5	84	14.0	17.5	Pinkut	73	-3.5	2.0
11	Babine Fulton	1	3.5	84	14.0	17.5	Fulton	73	3.5	2.0
12	Swan/Stephans/Club	3	-10.5	76	12.6	12.6	Swan+	72	-10.5	1.8
13	Bulkley/Maxan	2	-10.5	105	17.5	17.5	Bulkley-Morice	72	-10.5	2.5
14	Morice/Atna	2	-10.5	105	17.5	17.5	Bulkley-Morice	72	-10.5	2.5
15	Kitwanga	1	3.5	118	19.6	19.6	Kitwanga+	74	3.5	2.8
16	Zymoetz[4]	3	-17.5	59	9.8	9.8	Zymoetz	71	-17.5	1.4
17	Kalum	1	-3.5	105	17.5	17.5	Kalum-Bear	73	-3.5	2.5
18	Lakelse	1	-21.0	80	13.3	13.3	Lakelse+	64	-21	1.9
19	Alastair	1	-14.0	109	18.2	18.2	Alastair	71	-14	2.6
20	Johnston/Ecstall	1	-21.0	80	13.3	13.3	Lakelse+	64	-21	1.9

[1] Babine W Early includes sockeye spawing in non-enhanced tributaries to Babine Lake and in Onerka Lake.

[2] Babine W Middle includes the Tahlo/Morrison CU.

[3] Babine W Late includes the Nilkitkwa Lake CU.

[4] Zymoetz includes three sockeye lake CUs in the Zymoetz watershed (Mcdonell, Aldrich and Dennis).

Table 4. Initial annual estimates of the harvest of Skeena River sockeye by First Nations in FSC fisheries, 1982-2009.

Fishery	Group	1982	1983	1984	1985	1986	1987	1988	1989	1990	1991	1992	1993	1994	1995
Coastal	TTC	12,770	7,232	7,630	10,655	8,284	9,431	7,334	8,553	10,197	5,420		16,926	8,423	11,821
Marine to Kasiks	TTC	23,816	13,489	14,230	19,872	15,450	17,590	13,678	15,951	19,017	10,108		27,809	26,409	14,905
Coastal to Kasiks	TTC	36,586	20,721	21,860	30,526	23,734	27,021	21,012	24,504	29,214	15,528		44,735	34,832	26,726
Kasiks to Terrace	TTC	4,582	2,595	2,738	3,823	2,972	3,384	2,631	3,069	3,658	1,945		14,274	3,665	5,177
Terrace to Fiddler	TTC	11,653	6,600	6,963	9,723	7,560	8,606	6,693	7,805	9,305	4,946		9,019	14,418	9,757
Fiddler to Hazelton	GWWA	102,600	79,420	128,250	114,000	85,500	76,000	71,250	85,500	83,600	83,600	66,500	27,221	35,307	42,668
Hazelton to L. Babine	GWWA	5,400	4,180	6,750	6,000	4,500	4,000	3,750	4,500	4,400	4,400	3,500		1,858	2,246
Babine below Fence	GWWA												13,448		6,439
Babine Fence	LBN														
Babine Lake	LB/YECH	42,000	20,000	12,100	16,000	4,050		25,000	22,000	27,008	15,650	33,093	68,250	32,300	18,491
Pinkut Terminal	LBN														
Fulton Terminal	LBN														
Bulkley-Morice	GWWA	4,500	6,450		4,000	22,450	20,296	4,250	1,450		13,000	15,138	11,408	12,629	23,912
Sustut	TAKLA													1,302	
Total		207,320	139,966	178,660	184,072	150,766	139,307	134,586	148,828	157,185	139,069	118,231	188,355	136,311	135,416

Fishery	Group	1996	1997	1998	1999	2000	2001	2002	2003	2004	2005	2006	2007	2008	2009
Coastal	TTC	7,444	4,090	21,124	3,073	37,157		29,000	3,123	4,356	10,376	9,607	21,685	27,134	9,100
Marine to Kasiks	TTC	31,951	37,839	24,105	14,644	27,600		20,000		4,840	2,507		17,022	5,428	
Coastal to Kasiks	TTC	39,395	41,929	45,229	17,717	64,757		49,000	3,123	9,196	12,883	9,607	38,707	32,562	9,100
Kasiks to Terrace	TTC	5,927	4,656	1,951	2,294	1,544		4,905	6,075	7,056	4,360	5,803	4,168	5,966	10,763
Terrace to Fiddler	TTC	15,744	12,909	15,209	10,131	13,245	13,479	13,680	11,337	12,550	9,098	9,749	9,338	8,535	9,465
Fiddler to Hazelton	GWWA	21,058	32,880	50,369	51,854	58,444	49,531	56,258	60,876	66,295	64,144	68,859	26,306	63,494	35,946
Hazelton to L. Babine	GWWA	1,109	1,731	2,651	2,730	3,076	2,487	2,879	3,204	3,489	3,376	3,624	1,385	3,342	1,892
Babine below Fence	GWWA	2,802	1,637	195	3,366	2,658	5,000	1,091	533	333	1,273		16,643		107
Babine Fence	LBN														
Babine Lake	LB/YECH	39,422	13,699	9,744	23,220	23,300	24,080	24,785	32,000	31,441	33,117	38,600	36,070	48,901	43,957
Pinkut Terminal	LBN														
Fulton Terminal	LBN														
Bulkley-Morice	GWWA	14,453	15,512	3,674	675	1,905	1,289	331	456	278	197	2,085	219	2,391	1,644
Sustut	TAKLA	559	513	768	868	1,050	470	811	1,954	567	862	632	419	526	992
Total		140,469	125,466	129,790	112,855	169,979	96,336	153,740	119,558	131,205	129,310	138,959	133,254	165,717	113,866

1) Coastal = LaxKwalaams+Metlakatla (No Kitkatla+Hartley Bay) catch data
2) Marine to Kasiks = Prince Rupert catch data
3) Kasiks to Terrace = Kitsumkalum catch data
4) Terrace to Fiddler = Kitselas catch data
5) Fiddler to Hazelton = 0.95 * Skeena (e.g. Hazelton) catch data based on comments provided by J. Steward regarding location of catch + Gitanyow
6) Hazelton to Lower Babine = 0.05 * Skeena (e.g. Hazelton) catch data based on comments provided by J.Steward regrading location of catch
7) Kitsegass (L. Babine) = Babine catch data
8) All Babine Lake = sum of Nat'oo'ten catch data
9) Sustut = Takla catch data
10) 1982-1992 GWWA Kitsegass catch data are included in the Hazelton to Lower Babine catch estimates
11) 1982-2000 Moricetown data are the actual reported catch figures
12) Note: Some missing 1992-1982 catch by fishery calculated as area-specific IFF total catch*Prop. IFF catch for 1993-2000
These numbers are of unknown accuracy: all have been interpolated from Kerra Hoyseth's (2000) DFO review.USE WITH CAUTION

Table 5. Revised annual estimates of the harvest of Skeena River sockeye by First Nations in FSC fisheries, 1982-2009.

Fishery	Group	1982	1983	1984	1985	1986	1987	1988	1989	1990	1991	1992	1993	1994	1995	Average	%
Coastal	TTC	12,770	7,232	7,630	10,655	8,284	9,431	7,334	8,553	10,197	5,420	11,173	16,926	8,423	11,821	9,703	5.5%
Marine to Kasiks	TTC	23,816	13,489	14,230	19,872	15,450	17,590	13,678	15,951	19,017	10,108	18,959	27,809	26,409	14,905	17,949	10.1%
Coastal to Kasiks[1]	TTC	36,586	20,721	21,860	30,526	23,734	27,021	21,012	24,504	29,214	15,528	30,132	44,735	34,832	26,726	27,652	15.6%
Kasiks to Terrace[2]	Kalum	8,993	5,093	5,373	7,503	5,834	6,642	5,165	6,023	7,181	3,817	9,045	14,274	7,194	10,161	7,307	4.1%
Terrace to Fiddler	Kitselas	11,653	6,600	6,963	9,723	7,560	8,606	6,693	7,805	9,305	4,946	6,982	9,019	14,418	9,757	8,573	4.8%
Fiddler to Hazelton[3]	GWWA	52,107	40,335	65,134	57,897	43,423	38,598	36,186	43,423	42,458	42,458	33,773	13,133	17,931	21,670	39,181	22.1%
Hazelton to L. Babine[3]	GWWA	55,893	43,265	69,866	62,103	46,577	41,402	38,814	46,577	45,542	45,542	36,227	14,088	19,234	23,244	42,027	23.7%
Babine below Fence	GWWA												13,448		6,439	9,944	5.6%
Babine Fence	LBN																
Babine Lake[4]	LB/YECH	42,000	20,000	20,500	17,500	23,500	20,296	25,000	22,000	22,000	20,800	73,789	68,250	32,300	18,491	30,459	17.2%
Pinkut Terminal	LBN																
Fulton Terminal	LBN																
Bulkley-Morice[5]	GWWA	4,500	6,450	5,225	4,000	22,450	20,296	4,250	1,450	7,225	13,000	15,138	11,408	12,629	23,912	10,852	6.1%
Sustut	TAKLA													1,302		1,302	0.7%
Total		211,731	142,464	194,921	189,253	173,078	162,861	137,119	151,782	162,924	146,091	205,087	188,355	139,840	140,400	177,297	

Fishery	Group	1996	1997	1998	1999	2000	2001	2002	2003	2004	2005	2006	2007	2008	2009	Average	%
Coastal	TTC	7,444	4,090	21,124	3,073	37,157	14,560	29,000	17,918	18,783	18,156	21,351	15,300	20,014	17,617	17,542	11.8%
Marine to Kasiks	TTC	31,951	37,839	24,105	14,644	27,600	16,364	20,000	20,138	21,110	20,406	23,996	17,196	22,494	19,799	22,689	15.3%
Coastal to Kasiks[1]	TTC	39,395	41,929	45,229	17,717	64,757	30,923	49,000	38,056	39,894	38,562	45,348	32,496	42,507	37,416	40,231	27.0%
Kasiks to Terrace[2]	Kalum	11,633	9,139	3,829	4,503	3,031	6,329	9,627	11,924	13,849	8,558	11,390	8,181	11,710	10,763	8,890	6.0%
Terrace to Fiddler	Kitselas	15,744	12,909	15,209	10,131	13,245	13,479	13,680	11,337	12,550	9,098	9,749	9,338	8,535	9,465	11,748	7.9%
Fiddler to Hazelton[3]	GWWA	10,695	16,699	25,581	26,336	39,178	21,594	31,536	22,618	25,931	22,040	36,251	14,825	27,616	14,882	23,984	16.1%
Hazelton to L. Babine[3]	GWWA	11,472	17,912	27,439	28,248	16,915	21,943	22,311	35,086	36,556	42,304	39,409	13,280	29,788	17,509	25,727	17.3%
Babine below Fence	GWWA	2,802	1,637	195	3,366	2,658	5,000	1,091	533	333	1,273		16,643		107	2,970	2.0%
Babine Fence	LBN																
Babine Lake[4]	LB/YECH	39,422	13,699	9,744	23,220	23,300	24,080	24,785	32,000	31,441	33,117	38,600	36,070	48,901	58,597	31,213	21.0%
Pinkut Terminal	LBN																
Fulton Terminal	LBN																
Bulkley-Morice	GWWA	14,453	15,512	3,674	675	1,905	1,289	331	456	278	197	2,085	219	2,391	1,644	3,222	2.2%
Sustut	TAKLA	559	513	768	868	1,050	470	811	1,954	567	862	632	419	526	992	785	0.5%
Total[6]		146,175	129,949	131,668	115,064	166,038	125,108	153,172	153,965	161,400	156,010	183,464	131,471	171,974	151,375	148,770	

[1] Coastal to Kasiks catches for 2001 and 2003-09 were estimated by multiplying the adjusted total FSC catch by the average portion (24.7%) that Coastal to Kasiks catches represented of the reported catch for 1996-2009.

[2] Kasiks to Terrace catch was expand by 1.96 for all years except 1993 and 2009, based on results from TRM studies 2007-08.

Reported (1982-95)	Kalum	4,582	2,595	2,738	3,823	2,972	3,384	2,631	3,069	3,658	1,945		14,274	3,665	5,177	
Reported (1995-09)	Kalum	5,927	4,656	1,951	2,294	1,544		4,905	6,075	7,056	4,360	5,803	4,168	5,966	10,763	
TRM Est. (2007-08)													9476	9856		
Expansion Factor	**1.96**												2.27	1.65		

[3] Revised catch estimates for Fiddler-Hazelton and Hazelton-L.Babine fisheries (dark green shaded value are from SFC)

		1996	1997	1998	1999	2000	2001	2002	2003	2004	2005	2006	2007	2008	2009	
Fiddler to Hazelton	GWWA	10,695	16,699	25,581	26,336	39,178	21,594	31,536	22,618	25,931	22,040	36,251	14,825	27,616	14,882	25,647
Hazelton to L. Babine	GWWA	11,472	17,912	27,439	28,248	16,915	21,943	22,311	35,086	36,556	42,304	39,409	13,280	29,788	17,509	27,510
Total		22,167	34,611	53,020	54,584	56,093	43,538	53,847	57,705	62,488	64,344	75,661	28,105	57,404	32,391	53,157
% Fiddler-Hazelton (New)						70%	50%	59%	39%	41%	34%	48%	53%	48%	46%	48%

		1982	1983	1984	1985	1986	1987	1988	1989	1990	1991	1992	1993	1994	1995	
Fiddler to Hazelton	GWWA	52,107	40,335	65,134	57,897	43,423	38,598	36,186	43,423	42,458	42,458	33,773	13,133	17,931	21,670	
Hazelton to L. Babine	GWWA	55,893	43,265	69,866	62,103	46,577	41,402	38,814	46,577	45,542	45,542	36,227	14,088	19,234	23,244	
Total		108,000	83,600	135,000	120,000	90,000	80,000	75,000	90,000	88,000	88,000	70,000	27,221	37,165	44,914	

[4] Yellow highlight values are revised FSC estimates from Steve Cox-Rogers' Babine stock assessment tables.

[5] Highlighted values are averages of adjacent FSC catch estimates.

[6] Total catch estimates for 2001 and 2003-09 were estimated by expanding the total reported catch for fisheries above Kasiks by 1.32 (i.e. 1/0.753) because, on average, fisheries above Kasiks represent 75.3% of the total FSC harvest of Skeena sockeye.

Table 6. Annual estimates of the harvest of Skeena River sockeye by First Nations in ESSR and in-land "Demonstration" fisheries, 1993-2009.

Fishery	Group	1993	1994	1995	1996	1997	2000	2001	2006	2007	2008
Kasiks to Terrace[1]	TTC	3,919	4,009	14,720	60,016	5,093	14,998				
Terrace to Fiddler	TTC			2,878			7,770		81,790		
Fiddler to Hazelton	GWWA	24,202	21,249	79,943	165,551	91,554	139,345	38,957			67,289
Hazelton to L. Babine	GWWA	1,274	1,118		8,713	4,819	180,140	26,112			
Babine below Fence	GWWA			31,880			9,995	152,230	92,347		41,715
Babine Fence	LBN	104,340	15,900	45,000	312,812	86,459	56,203	138,240	138,180	13,777	104,585
Babine Lake	TTC										
Pinkut Terminal	LBN					36,982	65,821	32,220			37,388
Fulton Terminal	LBN			35,000		19,268	310,132	315,220	80,820		50,506
Bulkley-Morice	GWWA					1,208					
Total		133,735	42,276	209,421	547,092	245,383	784,404	702,979	393,137	13,777	301,483

[1] Terrace to Fiddler ESSR for 2006 is the DEMO fishery catch

Table 7. Harvest rate estimates for each Skeena sockeye sub-stock group in First Nation fisheries conducted in and adjacent to the Skeena watershed, 1982-2009.

Sub-Stock	1982	1983	1984	1985	1986	1987	1988	1989	1990	1991	1992	1993	1994	1995
Kluatan+	13.3	16.1	12.4	7.1	14.7	8.3	8.8	12.9	10.6	7.9	8.2	6.9	11.1	11.4
Motase	11.6	10.6	13.3	7.0	13.9	7.9	6.7	9.5	11.4	7.8	7.7	6.1	9.1	8.0
Sustut+	12.3	12.5	12.7	6.9	14.1	8.0	7.4	10.9	10.8	7.8	7.9	6.4	32.1	9.5
Bear+	12.3	12.5	12.7	6.9	14.1	8.0	7.4	10.9	10.8	7.8	7.9	6.4	10.0	9.5
Slamgeesh	12.3	12.5	12.7	6.9	14.1	8.0	7.4	10.9	10.8	7.8	7.9	6.4	10.0	9.5
Sicintine	12.3	12.5	12.7	6.9	14.1	8.0	7.4	10.9	10.8	7.8	7.9	6.4	10.0	9.5
Babine-WE	17.3	18.4	13.9	7.9	18.0	9.9	10.8	15.3	12.5	9.7	15.0	19.5	17.1	17.8
Babine-WM	15.7	14.8	14.4	7.7	17.1	9.5	9.2	12.9	12.8	9.4	13.9	17.3	14.8	15.1
Babine-WL	13.4	10.0	15.4	7.7	16.1	8.9	7.1	9.4	13.8	9.2	12.2	14.1	11.6	10.9
Babine-P	15.7	14.8	14.4	7.7	17.1	9.5	9.2	12.9	12.8	9.4	13.9	17.3	14.8	15.1
Babine-F	14.5	12.0	14.9	7.7	16.5	9.2	8.1	11.0	13.3	9.3	13.0	15.5	13.0	15.9
Swan+	8.6	9.9	7.1	4.3	8.8	5.3	5.1	7.8	6.6	4.4	5.3	5.8	8.9	9.2
Bulkley+ [1]	42.1	46.6	44.8	44.0	76.5	69.2	59.5	17.2	37.5	16.6	24.3	23.6	24.0	30.9
Morice+ [1]	42.1	46.6	44.8	44.0	76.5	69.2	59.5	17.2	37.5	16.6	24.3	23.6	24.0	30.9
Kitwanga	5.9	5.3	5.3	3.2	6.4	4.0	3.3	4.6	5.6	3.1	4.1	4.4	6.1	4.9
Zymoetz	3.6	3.1	1.6	1.5	3.0	2.2	2.1	3.2	2.4	1.2	2.5	3.4	4.5	3.1
Kalum	3.2	2.4	2.0	1.5	3.0	2.1	1.7	2.4	2.8	1.2	2.5	3.1	3.6	2.4
Lakelse	3.3	2.7	1.5	1.3	2.8	2.1	2.0	2.9	2.2	1.1	2.4	3.3	4.1	3.1
Alastair	2.5	2.0	1.4	1.0	2.2	1.6	1.4	2.1	1.8	0.9	1.8	2.1	2.8	1.3
Johnston	0.9	0.7	0.4	0.3	0.7	0.5	0.5	0.8	0.5	0.3	0.5	0.7	1.0	0.5
Total	14.6	12.8	14.8	7.7	17.9	10.2	8.4	11.1	13.4	9.5	12.9	15.4	13.6	15.7

	1996	1997	1998	1999	2000	2001	2002	2003	2004	2005	2006	2007	2008	2009
Kluatan+	13.0	15.7	19.6	12.2	27.0	6.7	15.7	8.3	13.6	17.4	14.1	5.5	13.7	10.4
Motase	12.6	14.8	17.6	10.7	20.3	7.0	16.5	8.2	11.6	13.6	12.2	6.2	13.3	9.7
Sustut+	23.3	24.9	31.5	27.2	38.8	16.6	30.2	34.0	35.2	50.9	51.2	19.4	75.1	68.3
Bear+	12.7	15.3	18.4	11.4	22.7	6.8	16.0	8.2	12.3	14.9	13.0	5.8	13.5	10.0
Slamgeesh	12.7	15.3	18.4	11.4	22.7	6.8	16.0	8.2	12.3	14.9	13.0	5.8	13.5	10.0
Sicintine	12.7	15.3	18.4	11.4	22.7	6.8	16.0	8.2	12.3	14.9	13.0	5.8	13.5	10.0
Babine-WE	29.5	25.3	21.6	16.9	31.3	20.7	19.8	11.2	17.7	23.1	32.4	11.2	29.4	19.7
Babine-WM	28.0	24.0	20.2	15.4	27.3	21.0	19.9	10.8	15.9	19.9	29.7	11.3	28.5	17.9
Babine-WL	26.5	21.8	18.2	13.1	20.5	22.4	20.2	10.2	13.3	15.7	25.1	12.1	27.1	15.4
Babine-P	28.0	36.5	20.2	15.4	44.5	33.3	19.9	10.8	15.9	19.9	30.2	11.3	38.6	17.9
Babine-F	27.1	25.3	19.0	14.2	42.5	50.9	20.0	10.5	14.4	17.5	34.2	11.7	33.1	16.5
Swan+	12.0	13.8	14.2	7.9	15.1	4.6	12.6	5.5	8.9	9.8	11.2	4.3	11.5	8.0
Bulkley+	22.8	33.7	31.9	9.8	33.3	15.1	14.4	7.5	10.6	11.2	20.4	5.1	20.5	13.5
Morice+	22.8	33.7	31.9	9.8	33.3	15.1	14.4	7.5	10.6	11.2	20.4	5.1	20.5	13.5
Kitwanga	8.5	9.2	11.5	5.6	9.1	3.5	11.4	4.9	7.1	7.3	9.1	4.5	7.9	6.9
Zymoetz	4.3	4.2	7.7	3.3	4.5	1.5	6.6	3.2	5.3	6.0	5.4	2.4	3.4	5.2
Kalum	4.2	4.0	6.9	2.7	3.8	1.4	7.1	3.1	4.7	5.2	3.1	2.8	3.6	4.9
Lakelse	4.1	3.8	6.8	2.9	4.5	1.2	6.0	2.9	5.0	5.7	3.6	2.3	3.3	5.0
Alastair	1.3	2.8	6.2	2.1	3.2	1.0	5.3	2.1	3.3	4.3	2.6	1.9	2.6	3.6
Johnston	0.4	0.8	2.0	0.7	1.2	0.3	1.5	0.7	1.1	1.5	0.9	0.5	0.7	1.1
Total	26.2	26.3	18.7	13.7	38.9	33.8	19.2	10.1	13.9	17.0	31.3	11.0	32.6	15.9

[1] Harvest rate estimates for Bulkley+ and Morice+ sub-stock groups are likely biased high due to underestimated escapement from 1982-88.

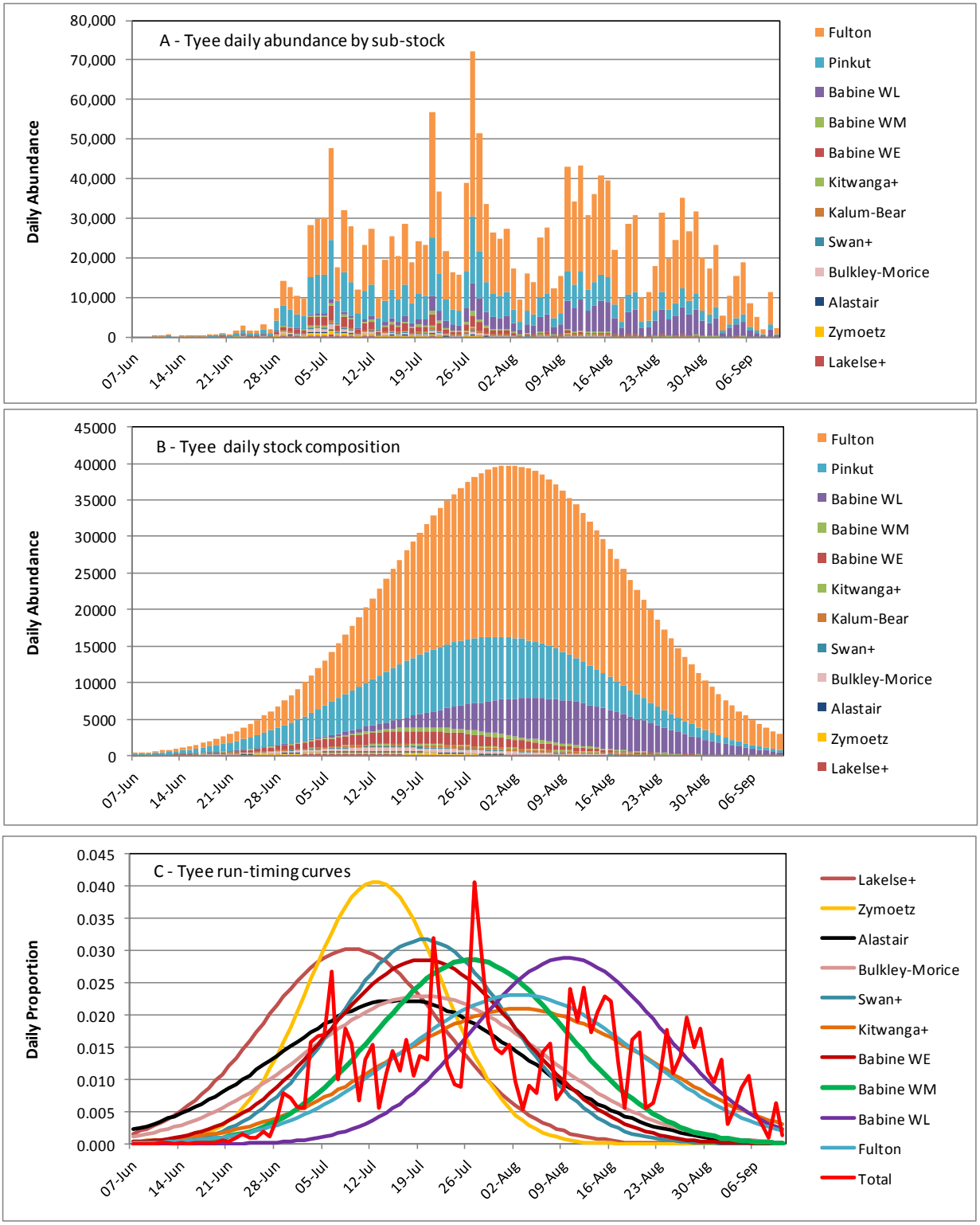

Figure 1. Estimated Tyee daily abundance, stock composition and run-timing curves for Skeena sockeye stocks in 2006 (late run-timing year).

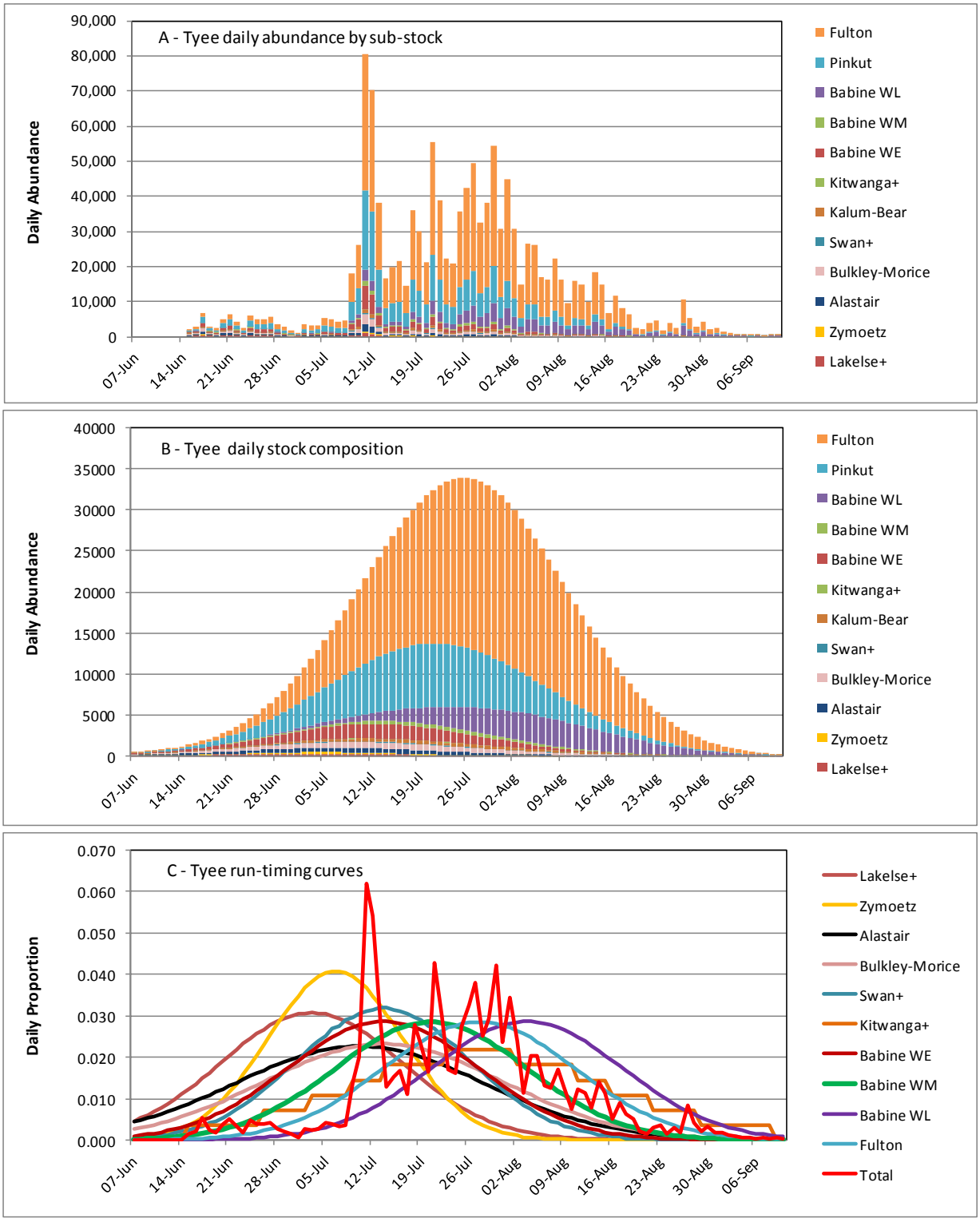

Figure 2. Estimated Tyee daily abundance, stock composition and run-timing curves for Skeena sockeye stocks in 2007 (average run-timing year).

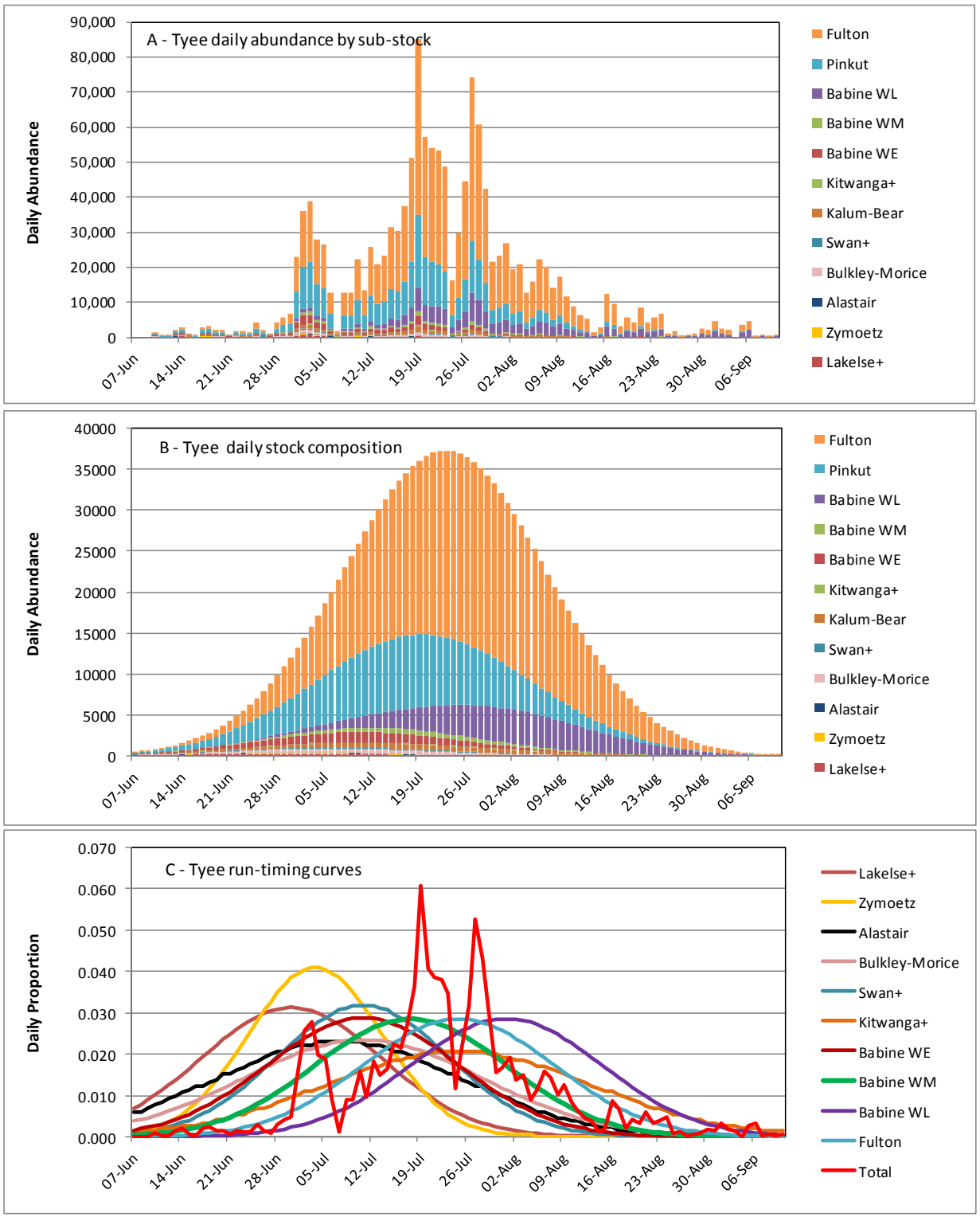

Figure 3. Estimated Tyee daily abundance, stock composition and run-timing curves for Skeena sockeye stocks in 2008 (early run-timing year).

APPENDIX A

Summary of catch and escapement results for 1982-2009

Table A-1 provides a summary of the annual escapement for each sub-stock and a comparison of the total output escapement and input escapement from Table 2. The difference between these two total "escapement" values is because the Table 2 totals include catches in terminal fisheries for each of the Babine sub-stocks whereas the output escapement estimates do not include these catches and the escapement estimates for non-Babine stock used to derive the initial abundance at Tyee are larger than the sum of the escapement estimates for non-Babine CUs included in Table 2. Consequently, in years without major terminal fisheries for Babine stocks, the model output escapement is larger than the input totals from Table 2 and in years with major terminal fisheries, the model output escapement estimates are smaller than the input totals from Table 2.

Table A-2 provides a summary of the annual estimates of the river entry abundance at Tyee for each sub-stock derived by combining the stock proportions from Table 2, the relative run-timing data for each sub-stock and the daily estimates of total sockeye abundance at Tyee. The total daily sockeye abundance at Tyee was computed by combining the total river entry escapement estimates (as used in the Northern Boundary run reconstruction analyses) with the daily Tyee test fishery CPUE data. The "Tyee Estimates" at the bottom of Table A-2 are provided to show that the sum of the sub-stock specific river-entry escapement estimates were always within a few fish of the input values for the total "Tyee Estimate".

Table A-3 provides the in-river catch by sub-stock calculated as the difference between the sockeye abundance at Tyee (Table A-2) and the escapement (Table A-1). The total annual catch estimated for all Skeena stocks is compared with the input catch at the bottom of Table A-3 to show that the model accounts for essentially all the catch in most years. The only year when the unaccounted for catch exceeded 4% of the total catch was 2001. Most of the unaccounted for catch in 2001 was associated with a very large terminal ESSR fishery for the Fulton enhanced sub-stock, where the method used to compute the terminal harvest rate was biased low.

Table A-1. Spawn ground escapement data from Reconstruction. Input File = "Excape.xls"

Stock	1982	1983	1984	1985	1986	1987	1988	1989	1990	1991	1992	1993	1994	1995
Kluatan+	1010	1031	846	1180	945	896	1233	1067	1093	956	963	1131	962	976
Motase	529	502	520	512	530	1592	99	407	62	320	534	415	251	283
Sustut+	4229	4256	4062	4306	4168	4050	4439	4252	4399	4127	4371	4331	4075	4025
Bear+	1001	981	911	2916	3117	11053	8013	1772	2507	7791	5570	6062	8209	10582
Slamgeesh	1057	1064	1015	1076	1042	1012	1110	1063	1100	1032	1093	1083	1019	1006
Sicintine	1057	1064	1015	1076	1042	1012	1110	1063	1100	1032	1093	1083	1019	1006
Babine-WE	93150	27640	23336	87165	25312	34369	51822	19448	23347	56769	50383	16509	23378	72408
Babine-WM	5277	9555	8302	18887	4139	16285	26506	8311	8254	26550	9660	26787	7778	6943
Babine-WL	164045	94357	211314	605793	175214	257960	210675	126847	184331	471454	574097	538304	126233	84959
Babine-P	246698	204123	365434	638276	225397	675766	418426	248142	223691	456526	230913	515848	206767	569578
Babine-F	650289	549819	463949	838465	288108	356490	691855	736072	549805	216164	366783	518946	649759	970039
Swan+	9351	10779	10298	10999	11226	20042	32930	9749	12459	15390	23415	19578	12910	8091
Bulkley+	675	655	533	712	260	301	548	1015	764	866	793	928	822	761
Morice+	7335	9795	6180	5871	7499	8673	3493	13548	15260	84855	57539	53962	60159	74753
Kitwanga	2133	2820	3842	4807	4220	3366	2482	1805	1069	1165	1102	1051	1039	645
Zymoetz	2377	12074	801	1532	6139	4951	6313	4357	3510	1182	10123	19930	10871	11883
Kalum	873	698	839	1832	1792	3287	2247	3255	4803	4524	12039	10874	16123	15414
Lakelse	33684	24365	6674	52048	8598	4273	16566	9786	3264	12465	9117	17785	3141	28757
Alastair	10474	15684	6950	10384	21664	9046	18018	18009	11975	21645	15412	17958	13262	17854
Johnston	574	905	497	1305	343	492	1128	391	561	569	1770	4476	4916	6557
Total	**1235818**	**972167**	**1117318**	**2289142**	**790755**	**1414916**	**1499013**	**1210359**	**1053354**	**1385382**	**1376770**	**1777041**	**1152693**	**1886520**
Input escapement	**1206037**	**960210**	**1102333**	**2225396**	**771592**	**1385995**	**1484109**	**1193690**	**1033586**	**1334424**	**1376332**	**1872830**	**1189565**	**1913146**
Difference	**29781**	**11957**	**14985**	**63746**	**19163**	**28921**	**14904**	**16669**	**19768**	**50958**	**438**	**-95789**	**-36872**	**-26626**

Table A-2. Incoming abundance to Tyee from reconstruction. Input File = "Stock_Calc_Entry.xls"

Stock	1982	1983	1984	1985	1986	1987	1988	1989	1990	1991	1992	1993	1994	1995
Kluatan+	1166	1228	966	1270	1108	977	1352	1225	1222	1039	1049	1215	1082	1101
Motase	598	561	599	551	616	1728	106	450	69	347	578	441	277	308
Sustut+	4823	4863	4654	4627	4852	4403	4791	4771	4932	4475	4746	4627	6002	4449
Bear+	1142	1121	1044	3133	3628	12017	8650	1989	2810	8448	6048	6476	9121	11697
Slamgeesh	1206	1216	1164	1157	1213	1101	1198	1193	1233	1119	1186	1157	1132	1112
Sicintine	1206	1216	1164	1157	1213	1101	1198	1193	1233	1119	1186	1157	1132	1112
Babine-WE	112663	33880	27115	94625	30851	38146	58066	22966	26691	62883	59250	20520	28209	88127
Babine-WM	6263	11216	9698	20468	4991	17992	29205	9547	9466	29313	11218	32406	9134	8177
Babine-WL	189398	104824	249852	656149	208859	283250	226827	140026	213953	519074	653815	626787	142778	95305
Babine-P	292807	239610	426899	691693	271816	746592	461038	285049	256546	504043	268143	624062	242816	670745
Babine-F	760177	625121	545115	908041	345172	392494	752519	826980	633974	238215	421628	614367	747175	1153392
Swan+	10229	11963	11088	11498	12309	21174	34708	10574	13344	16104	24733	20794	14164	8914
Bulkley+	1166	1228	966	1270	1108	977	1352	1225	1222	1039	1049	1215	1082	1101
Morice+	12678	18358	11205	10477	31937	28130	8619	16361	24416	101791	76054	70665	79171	108250
Kitwanga	2266	2977	4058	4967	4508	3505	2567	1893	1132	1203	1148	1100	1106	678
Zymoetz	2465	12457	814	1554	6330	5064	6446	4500	3596	1197	10380	20632	11386	12266
Kalum	902	715	856	1860	1848	3358	2286	3336	4940	4581	12342	11220	16727	15795
Lakelse	34850	25040	6775	52738	8847	4363	16900	10083	3337	12602	9338	18393	3277	29689
Alastair	10746	15999	7046	10494	22157	9195	18275	18389	12201	21840	15689	18349	13642	18094
Johnston	579	912	499	1309	346	494	1134	394	564	571	1780	4508	4964	6589
Total	**1447330**	**1114505**	**1311577**	**2479038**	**963709**	**1576061**	**1637237**	**1362144**	**1216881**	**1531003**	**1581360**	**2100091**	**1334377**	**2236901**
Tyee estimate	**1447331**	**1114505**	**1311576**	**2479034**	**963708**	**1576062**	**1637237**	**1362144**	**1216885**	**1531001**	**1581359**	**2100091**	**1334376**	**2236902**

Table A-3. Catch calculated as the difference between Tyee run size and total escapement for each CU.

Stock	1982	1983	1984	1985	1986	1987	1988	1989	1990	1991	1992	1993	1994	1995
Kluatan+	156	197	120	90	163	81	119	158	129	83	86	84	120	125
Motase	69	59	79	39	86	136	7	43	7	27	44	26	26	25
Sustut+	594	607	592	321	684	353	352	519	533	348	375	296	1927	424
Bear+	141	140	133	217	511	964	637	217	303	657	478	414	912	1115
Slamgeesh	149	152	149	81	171	89	88	130	133	87	93	74	113	106
Sicintine	149	152	149	81	171	89	88	130	133	87	93	74	113	106
Babine-WE	19513	6240	3779	7460	5539	3777	6244	3518	3344	6114	8867	4011	4831	15719
Babine-WM	986	1661	1396	1581	852	1707	2699	1236	1212	2763	1558	5619	1356	1234
Babine-WL	25353	10467	38538	50356	33645	25290	16152	13179	29622	47620	79718	88483	16545	10346
Babine-P	46109	35487	61465	53417	46419	70826	42612	36907	32855	47517	37230	108214	36049	101167
Babine-F	109888	75302	81166	69576	57064	36004	60664	90908	84169	22051	54845	95421	97416	183353
Swan+	878	1184	790	499	1083	1132	1778	825	885	714	1318	1216	1254	823
Bulkley+	491	573	433	558	848	676	804	210	458	173	256	287	260	340
Morice+	5343	8563	5025	4606	24438	19457	5126	2813	9156	16936	18515	16703	19012	33497
Kitwanga	133	157	216	160	288	139	85	88	63	38	46	49	67	33
Zymoetz	88	383	13	22	191	113	133	143	86	15	257	702	515	383
Kalum	29	17	17	28	56	71	39	81	137	57	303	346	604	381
Lakelse	1166	675	101	690	249	90	334	297	73	137	221	608	136	932
Alastair	272	315	96	110	493	149	257	380	226	195	277	391	380	240
Johnston	5	7	2	4	3	2	6	3	3	2	10	32	48	32
Total	**211512**	**142338**	**194259**	**189896**	**172954**	**161145**	**138224**	**151785**	**163527**	**145621**	**204590**	**323050**	**181684**	**350381**
Catch input	**211731**	**142464**	**194921**	**189253**	**173078**	**162861**	**137119**	**151782**	**162924**	**146091**	**205087**	**322090**	**182116**	**349821**
Difference	-219	-126	-662	643	-124	-1716	1105	3	603	-470	-497	960	-432	560

Table A-1. Spawn ground escapement estimates data from reconstruction. (continued)

Stock	1996	1997	1998	1999	2000	2001	2002	2003	2004	2005	2006	2007	2008	2009
Kluatan+	1156	961	1068	917	1123	1254	1043	1100	1164	1073	1086	1077	778	1021
Motase	105	215	493	552	388	204	101	2100	598	293	126	318	112	453
Sustut+	4272	3837	4048	4194	4288	4442	4337	5454	1753	1131	828	2728	200	609
Bear+	6729	3660	1598	9641	5678	4828	973	12293	3277	2405	2919	3533	8104	7659
Slamgeesh	1067	959	1012	1049	1017	949	432	470	320	208	339	404	142	182
Sicintine	1067	959	1012	1049	1072	1111	1084	1092	1093	962	1024	1104	945	1129
Babine-WE	56493	80081	45729	57353	90068	248408	31422	60495	44882	23786	35221	55110	18041	19836
Babine-WM	10372	37861	16968	24201	30331	80187	27005	91635	56928	30089	13672	13897	13700	20743
Babine-WL	124320	119648	93064	173265	152201	483409	118942	97899	218137	169348	126887	95947	101408	93930
Babine-P	566724	161694	98967	141298	214495	165599	104573	238811	112686	124696	283534	249188	177470	161632
Babine-F	975349	501319	237151	235636	883776	447866	297212	717696	450660	323973	707662	644850	584144	391644
Swan+	8686	5946	6861	4287	7569	9415	3055	5975	6954	2762	5126	2372	3519	3929
Bulkley+	1026	756	905	943	1026	1141	1059	1110	1205	1153	1006	1082	717	985
Morice+	104188	51207	14864	30690	8496	11166	16096	24100	20172	19838	19747	31059	15417	25590
Kitwanga	293	274	300	301	252	241	1047	3721	1381	1128	5918	262	1579	3452
Zymoetz	14090	12378	14656	9498	16569	15909	9668	11677	9364	8519	8620	4193	181	3659
Kalum	6357	11143	12800	13812	18453	10010	12815	32483	16579	10428	11410	13821	16344	22175
Lakelse	34371	3924	7548	6039	12684	14421	8565	9137	8670	8937	4946	7290	3210	7323
Alastair	33439	26132	13793	1913	9547	14814	4505	31349	26644	17090	5919	24198	884	22435
Johnston	7654	4770	8986	6973	2721	7229	2405	6324	3680	3548	3352	2683	1432	2409
Total	**1957758**	**1027724**	**581823**	**723611**	**1461754**	**1522603**	**646339**	**1354921**	**986147**	**751367**	**1239342**	**1155116**	**948327**	**790795**
Input escapement	**2179432**	**1207711**	**582814**	**700197**	**1896806**	**2055757**	**653620**	**1298738**	**999714**	**774542**	**1452455**	**1137899**	**1144813**	**768145**
Difference	**-221674**	**-179987**	**-991**	**23414**	**-435052**	**-533154**	**-7281**	**56183**	**-13567**	**-23175**	**-213113**	**17217**	**-196486**	**22650**

Table A-2. Incoming abundance to Tyee from reconstruction. (continued)

Stock	1996	1997	1998	1999	2000	2001	2002	2003	2004	2005	2006	2007	2008	2009
Kluatan+	1328	1140	1328	1045	1538	1343	1237	1200	1347	1298	1264	1140	902	1139
Motase	120	253	599	619	486	220	121	2286	676	339	143	339	130	501
Sustut+	5568	5108	5914	5760	7007	5329	6213	8263	2705	2303	1696	3384	806	1921
Bear+	7705	4319	1958	10881	7350	5183	1159	13386	3736	2825	3356	3749	9367	8508
Slamgeesh	1221	1132	1240	1183	1317	1019	514	512	365	244	390	429	164	202
Sicintine	1221	1132	1240	1183	1387	1192	1291	1190	1246	1130	1178	1172	1092	1254
Babine-WE	80092	107218	58348	69049	131101	313249	39158	68142	54510	30943	52136	62052	25562	24699
Babine-WM	14413	49794	21271	28603	41702	101532	33695	102693	67681	37578	19439	15674	19151	25278
Babine-WL	169208	153068	113741	199498	191335	622889	149066	109077	251509	200836	169509	109213	139080	110990
Babine-P	787490	254515	124068	167000	386236	248090	130481	267631	133971	155732	406145	281069	288916	196964
Babine-F	1337647	670740	292907	274534	1537370	912814	371494	801489	526717	392760	1076110	730212	872835	469209
Swan+	9865	6898	7998	4657	8915	9874	3496	6326	7636	3064	5772	2480	3975	4269
Bulkley+	1328	1140	1328	1045	1538	1343	1237	1200	1347	1298	1264	1140	902	1139
Morice+	134925	77227	21817	34009	12735	13149	18809	26054	22564	22332	24805	32722	19397	29595
Kitwanga	321	302	339	319	278	250	1182	3912	1486	1217	6510	274	1715	3708
Zymoetz	14723	12916	15873	9821	17353	16149	10350	12058	9884	9062	9110	4295	188	3861
Kalum	6636	11611	13749	14195	19179	10156	13788	33536	17390	10997	11778	14225	16960	23326
Lakelse	35838	4080	8096	6218	13280	14599	9113	9414	9127	9480	5129	7459	3319	7706
Alastair	33871	26873	14698	1955	9858	14966	4756	32029	27545	17850	6075	24662	907	23267
Johnston	7684	4809	9173	7024	2753	7251	2441	6366	3721	3602	3382	2697	1442	2437
Total	**2651204**	**1394275**	**715685**	**838598**	**2392718**	**2300597**	**799601**	**1506764**	**1145163**	**904890**	**1805191**	**1298387**	**1406810**	**939973**
Tyee estimate	**2651205**	**1394274**	**715687**	**838599**	**2392717**	**2300597**	**799602**	**1506763**	**1145163**	**904893**	**1805191**	**1298389**	**1406809**	**939972**

Table A-3. Catch calculated as the difference between Tyee run size and total escapement for each CU. (continued)

Stock	1996	1997	1998	1999	2000	2001	2002	2003	2004	2005	2006	2007	2008	2009
Kluatan+	172	179	260	128	415	89	194	100	183	225	178	63	124	118
Motase	15	38	106	67	98	16	20	186	78	46	17	21	18	48
Sustut+	1296	1271	1866	1566	2719	887	1876	2809	952	1172	868	656	606	1312
Bear+	976	659	360	1240	1672	355	186	1093	459	420	437	216	1263	849
Slamgeesh	154	173	228	134	300	70	82	42	45	36	51	25	22	20
Sicintine	154	173	228	134	315	81	207	98	153	168	154	68	147	125
Babine-WE	23599	27137	12619	11696	41033	64841	7736	7647	9628	7157	16915	6942	7521	4863
Babine-WM	4041	11933	4303	4402	11371	21345	6690	11058	10753	7489	5767	1777	5451	4535
Babine-WL	44888	33420	20677	26233	39134	139480	30124	11178	33372	31488	42622	13266	37672	17060
Babine-P	220766	92821	25101	25702	171741	82491	25908	28820	21285	31036	122611	31881	111446	35332
Babine-F	362298	169421	55756	38898	653594	464948	74282	83793	76057	68787	368448	85362	288691	77565
Swan+	1179	952	1137	370	1346	459	441	351	682	302	646	108	456	340
Bulkley+	302	384	423	102	512	202	178	90	142	145	258	58	185	154
Morice+	30737	26020	6953	3319	4239	1983	2713	1954	2392	2494	5058	1663	3980	4005
Kitwanga	28	28	39	18	26	9	135	191	105	89	592	12	136	256
Zymoetz	633	538	1217	323	784	240	682	381	520	543	490	102	7	202
Kalum	279	468	949	383	726	146	973	1053	811	569	368	404	616	1151
Lakelse	1467	156	548	179	596	178	548	277	457	543	183	169	109	383
Alastair	432	741	905	42	311	152	251	680	901	760	156	464	23	832
Johnston	30	39	187	51	32	22	36	42	41	54	30	14	10	28
Total	**693446**	**366551**	**133862**	**114987**	**930964**	**777994**	**153262**	**151843**	**159016**	**153523**	**565849**	**143271**	**458483**	**149178**
Catch input	**693267**	**375332**	**133527**	**115064**	**950442**	**828087**	**153172**	**153965**	**161400**	**156010**	**576601**	**145248**	**473457**	**151375**
Difference	179	-8781	335	-77	-19478	-50093	90	-2122	-2384	-2487	-10752	-1977	-14974	-2197

APPENDIX B

Skeena Sockeye In-River Model Visual Basic Code

```
' This program implements the Skeena River Sockeye Run Reconstruction
' by Karl English, Bill Gazey and Cam Noble Version 3 - January 2013.
' The model reconstructs the 1982-specified sockeye salmon returns to the Skeena River using First Nation provided estimates for in-river harvests, Tyee specific escapement numbers and timing provided by Steve Cox Rogers and Karl English, and upstream migration rates.
' The model combines catch data for 11 fisheries with escapement data on a daily time step for 20 sockeye stocks.
‘Terminal runs, harvest rates, and catch are estimated for each stock.
' The reconstruction algorithm is based on Cave and Gazey 1994.
'
' The model currently has the following data limits:
'   12 fisheries
'   20 stocks
'   1982 = Year Start
'   2011 = Year Finish
'
Option Base 1 'Start indexing arrays at 1
'Input data workbook declarations.
Public Fisheries_WkBk As Workbook
'Contains weekly FSC sockeye catch by fishery, week and year (each year a tab) and the Tyee aggegate run by day and year
Public FisheriesESSR_WkBk As Workbook
'Contains weekly ESSR sockeye catch by fishery, week and year (each year a tab)
Public Entry_WkBk As Workbook
'Annual entry (escapement from marine fisheries)estimates for each stock.
Public Timing_WkBk As Workbook
 'The timing workbook:
                ' "FISHRES" has the names and residence times for each stock in the fisheries
                ' "TIMING" has the offsets and standard deviations (days) for each stock from the aggregate at Tyee
                ' "Timing_Mean" has the mean date of arrival for stock aggregate at Tyee for each year
                ' Note that 2010 and 2011 do not have estimates yet
'Output workbook declarations
Public Reconstruct_WkBk As Workbook             'Contains catch reconstruction for each year.
Public Catch_WkBk As Workbook                   'Annual terminal catch summary for each year.
Public Escape_WkBk As Workbook                   'Annual terminal escapement (spawning ground) size by stock.
Public ExploitRate_WkWb As Workbook             ' Contains 2 worksheets:
                                                '   "StockGroup" is the annual exploitation rate by stock group.
                                                '   "Stock" is the annual exploitation rate by stock.
Public Stock_Calc_Entry_WkBk As Workbook        'Annual entry at Tyee as estimated by run recon (CNoble Oct. 11)
Public FisheryHr_WkBk As Workbook               'Daily fishery harvest rates estimated each week
Public WeeklyHr_WkBk As Workbook                'Mean daily fishery harvest rate.
Public Unexpl_WkBk As Workbook                  'Weekly unxplained catch by fishery and year
' Constant Declarations
Public Const BaseYear As Integer = 1982         'First year of data
Public Const DaysInModel As Integer = 140       'Number model days -- starts June 1
Public Const NumFisheries As Integer = 12       'Number of fisheries
Public Const NumSGroups As Integer = 2          'Number of stock groups -- Babine and non-Babine
Public Const NumStocks As Integer = 20          'Number of fish stocks (CUs)
Public Const WeeksInModel As Integer = 21       'Number of weeks in model
Public Const NumSectors As Integer = 2          'Number of fishing sectors
'Input Workbooks data offsets
Public Const Timing_Stock_DataRow = 3
```

```
Public Const Timing_Stock_DataCol = 2

Public Const Fisheries_DataCol = 4          'Column for first fishery in Fisheries_WkBk (FSC) work sheet
Public Const Fisheries_DataRow = 2          'First row of weekly catch data in Fisheries_WkBk (FSC),
catch-year worksheet (CNoble Oct 11)
'Output Workbooks data offsets
Public Const Reconst_Datarow As Integer = 2     'First row to begin output of reconstruction data (CNoble Oct
11)
Public Const Reconst_CatchRow As Integer = NumFisheries + 3 'Row to output total catch of reconstruction data
Public Const Year_Datacol As Integer = 2        'First column to begin output of yearly related sheets
Public Const WkRun_Datarow As Integer = 3       'First row to begin output of daily run data (CNoble Oct 11)
Public Const FishHR_Datarow As Integer = 3      'First row to begin output of daily fishery harvest rate
Public Const FishHR_FishCol As Integer = 2      'Column number for first fishery in daily fishery harvest rate
workbook
'Shared Variables
Public StartYear As Integer             'Start year for this model run
Public EndYear As Integer               'End year for this model run
'Shared data structures
Public StockToStockGroup(NumStocks) As Integer          'contains the mapping for each stock to its stock
group
Public Residence(NumStocks, NumFisheries) As Integer    'contains the fishery residence times for each stock
Public Cum_ResTimes(NumStocks, NumFisheries) As Integer  'contains the cumulative fishery residence times
from
                                                        'escapement to each fishery for each stock
Public Spawn_Peak(NumStocks) As Double          'contains the peak spawing model day for each stock
Public Spawn_SD(NumStocks) As Double            'contains the run sd for each stock
Public Spawn_Entry(NumStocks) As Double         'contains entry size for each stock
Public Stock_Calc_Entry(NumStocks) As Double    'contains calculated entry size for each stock (CNoble
Oct 11)
Public Stock_Daily_Escape(NumStocks, DaysInModel) As Double     'total daily escapement for each stock
Public StockGroupYearCatch(NumSGroups) As Double        'contains the total catch for each stock group in a
given year
Public StockYearCatch(NumStocks) As Double              'contains the total catch for each stock in a given
year
Public StockGroupTrun(NumSGroups) As Double
Public StockTrun(NumStocks) As Double
Public RunTotal As Double
Public CatchBySector(NumStocks, NumSectors) As Double   'contains annual catch by sector
Public FisheryNames(NumFisheries) As String             'contains fishery names
Public WeeklyHR(NumFisheries, WeeksInModel) As Double   'contains weekly harvest rates by fishery (computed)
Public FisheryHR(NumFisheries, WeeksInModel) As Double  'contains daily harvest rates by fishery and week
Public FSC_Catch(NumFisheries, WeeksInModel) As Double
Public ESSR_Catch(NumFisheries, WeeksInModel) As Double
Public Total_Catch(NumFisheries) As Double              'KKE added 14 Oct 2012 for entry run size and terminal
fishery
Public TerminalHR(NumFisheries) As Double
Public Week_CatchDay(WeeksInModel) As Integer
Sub Read_Catch(Yr, Year_String)
Const Fisheries_row As Integer = 2
Const FisheriesDate_col As Integer = 1
Const FisheriesFirst_col As Integer = 4
Dim iweek As Integer, ifish As Integer
Dim curdate As Date, bdate As Date
Dim shtname As String
Dim wsheet As Worksheet
Erase Total_Catch
With Fisheries_WkBk.Worksheets(Year_String)
   For iweek = 1 To WeeksInModel
```

```
      curdate = .Cells(iweek - 1 + Fisheries_row, FisheriesDate_col)
      bdate = DateValue("June 1," & Format(Year(curdate)))
      Week_CatchDay(iweek) = DateDiff("d", bdate, curdate) - 5  'set to the start of the week
      For ifish = 1 To NumFisheries
         FSC_Catch(ifish, iweek) = .Cells(iweek - 1 + Fisheries_row, FisheriesFirst_col + ifish - 1).Value
         Total_Catch(ifish) = Total_Catch(ifish) + FSC_Catch(ifish, iweek)
      Next ifish
   Next iweek
End With
shtname = Year_String + "ESSR"
On Error Resume Next
Set wsheet = FisheriesESSR_WkBk.Worksheets(shtname)
If wsheet Is Nothing Then
   Erase ESSR_Catch
Else
   With FisheriesESSR_WkBk.Worksheets(shtname)
      For iweek = 1 To WeeksInModel
         For ifish = 1 To NumFisheries
            ESSR_Catch(ifish, iweek) = .Cells(iweek - 1 + Fisheries_row, FisheriesFirst_col + ifish - 1).Value
            Total_Catch(ifish) = Total_Catch(ifish) + ESSR_Catch(ifish, iweek)
         Next ifish
      Next iweek
   End With
End If
On Error GoTo 0
   End Sub
Sub Reconstruction()
Dim Year As Integer, YearCol As Integer, xYear As Integer
Dim Year_String As String 'Used to read the correct First Nation catch worksheet for the given year.
'Turn off display
Application.ScreenUpdating = False
xYear = 0
Call Init(xYear)
'initialize output rows
YearCol = Year_Datacol    'Start column to output data in year related worksheets

'do the run reconstruction for each year requested
For Year = StartYear To EndYear
   'Create the string for this year
   Year_String = Right(Str(Year), 4)
   'Read in the catch
   Call Read_Catch(Year, Year_String)
   'Pick up the duration and timing
   Call FishResSpawn(Year)
   'Calculate the daily escapement for each stock group
   Call Calc_Escape(Year)
   'Do the reconstruction
   Call Reconstruct(Year, Year_String)
   'Output the yearly results
   Call OutputData(Year, Year_String, YearCol)
   'Increment the output row pointers
   YearCol = YearCol + 1
Next Year
'Close the input workbooks
Fisheries_WkBk.Close (False)
FisheriesESSR_WkBk.Close (False)
Entry_WkBk.Close (False)
Timing_WkBk.Close (False)
```

```
'Save and close the output workbooks
Reconstruct_WkBk.Save
Reconstruct_WkBk.Close
Catch_WkBk.Save
Catch_WkBk.Close
Escape_WkBk.Save
Escape_WkBk.Close
Stock_Calc_Entry_WkBk.Save          'CNoble (Oct 11)
Stock_Calc_Entry_WkBk.Close         'CNoble (Oct 11)
ExploitRate_WkWb.Save
ExploitRate_WkWb.Close
FisheryHr_WkBk.Save
FisheryHr_WkBk.Close
WeeklyHr_WkBk.Save
WeeklyHr_WkBk.Close
Unexpl_WkBk.Save
Unexpl_WkBk.Close
'Quit Excel
'Application.Quit  'Comment out while debugging
End Sub
Sub Init(xYear)
'Reads in number of fishing days per week, and start/end years to run the model from the Control sheet
'Opens all data workbooks
'Prompts user for output workbook name, and creates output workbook
Dim Col_Num As Integer, StockGroup As Integer, Sheet As Integer, SheetCount As Integer, Year As Integer
'counters
Dim StockNames(NumStocks) As String     'Stock names
Dim StockGroupNames(NumSGroups) As String   'Stock group names
Dim FileNames(4) As String  'file names
Dim drivename As String, filepath As String, testfile As String
'Initialize the stock group names
StockGroupNames(1) = "Non Babine"
StockGroupNames(2) = "Babine"
'Prompt for the number of fishing days per week, start and end year
Worksheets("Control").Activate
With Range("Control")
   StartYear = .Cells(2, 1)
   EndYear = .Cells(2, 2)
End With
'Pick up file names
With Range("Files")
   For i = 1 To 4
      FileNames(i) = .Cells(i, 1)
   Next i
End With
'Get the path to this workbook, extract the drive, and explictly change to this workbook's drive and folder.
'This sets the default folder for the input and output data workbooks to that of the workbook containing this program
filepath = ThisWorkbook.Path
drivename = Left(filepath, 1)  'Extract the drive
ChDrive (drivename)
ChDir (filepath)
Set Fisheries_WkBk = Workbooks.Open(Filename:=FileNames(1), UpdateLinks:=0)
Set FisheriesESSR_WkBk = Workbooks.Open(Filename:=FileNames(2), UpdateLinks:=0)
Set Entry_WkBk = Workbooks.Open(Filename:=FileNames(4), UpdateLinks:=0)
Set Timing_WkBk = Workbooks.Open(Filename:=FileNames(3), UpdateLinks:=0)
'Read in the stock and fishery names from the Timing workbook
For Stock = 1 To NumStocks
```

```
   StockNames(Stock) = Timing_WkBk.Worksheets("FISHRES").Cells(Timing_Stock_DataRow + Stock - 1,
Timing_Stock_DataCol).Value
Next Stock
For Fishery = 1 To NumFisheries
   FisheryNames(Fishery) = Timing_WkBk.Worksheets("FISHRES").Cells(Timing_Stock_DataRow - 1,
Timing_Stock_DataCol + Fishery + 4)
Next Fishery
'Yearly run reconstruction workbook
testfile = Dir("Reconstruction.xls")
If Len(testfile) > 0 Then Kill ("Reconstruction.xls")
Set Reconstruct_WkBk = Workbooks.Add
Reconstruct_WkBk.SaveAs Filename:="Reconstruction.xls", FileFormat:=xlAddIn
With Reconstruct_WkBk
   Sheet = 1
   SheetCount = .Worksheets.Count
   For Year = StartYear To EndYear
      'Check if need to add a worksheet to hold this year's reconstruction output
      If Sheet > SheetCount Then
         .Worksheets.Add after:=Worksheets(Sheet - 1)
      End If
      .Worksheets(Sheet).Name = Right(Str(Year), 4) 'str function adds leading blank - remove by extract 4
rightmost characters
      'Initialize the column headings
      Col_Num = 1
      With .Worksheets(Sheet)
         .Cells(1, Col_Num).Value = "Fishery"
         Col_Num = Col_Num + 1
         For Stock = 1 To NumStocks
            .Cells(1, Col_Num).Value = StockNames(Stock) 'stock names start in 2nd column
            Col_Num = Col_Num + 1
         Next Stock
         .Cells(1, Col_Num).Value = "Total"
         'Fit the column width to the headings
         .Range(.Cells(1, 1), .Cells(1, Col_Num)).ColumnWidth = 13
         .Range(.Cells(1, 1), .Cells(1, Col_Num)).HorizontalAlignment = xlRight
      End With
      Sheet = Sheet + 1
   Next Year
End With
'catch summary workbook
testfile = Dir("Catch.xls")
If Len(testfile) > 0 Then Kill ("Catch.xls")
Set Catch_WkBk = Workbooks.Add
Catch_WkBk.SaveAs Filename:="Catch.xls", FileFormat:=xlAddIn   'Later => prompt for output filename
Catch_WkBk.Worksheets(1).Name = "Catch"
With Catch_WkBk.Worksheets("Catch")
   .Cells(1, 1).Value = "Stock Group"
   Row_Num = 3
   For StockGroup = 1 To NumSGroups
      .Cells(Row_Num, 1).Value = StockGroupNames(StockGroup) 'stock names start in 2nd row
      Row_Num = Row_Num + 1
   Next StockGroup
   .Cells(Row_Num, 1).Value = "Total"
   'Fit the column width to the headings
   .Columns(1).AutoFit
End With
'Initialize catch by sector worksheets
Catch_WkBk.Worksheets(2).Name = "FSCCatchByGroup"
```

```
With Catch_WkBk.Worksheets("FSCCatchByGroup")
   .Cells(1, 1).Value = "Stock Group"
   Row_Num = 3
   For StockGroup = 1 To NumSGroups
      .Cells(Row_Num, 1).Value = StockGroupNames(StockGroup) 'stock names start in 2nd row
      Row_Num = Row_Num + 1
   Next StockGroup
   .Cells(Row_Num, 1).Value = "Total"
   'Fit the column width to the headings
   .Columns(1).AutoFit
End With
Catch_WkBk.Worksheets(3).Name = "ESSRCatchByGroup"
With Catch_WkBk.Worksheets("ESSRCatchByGroup")
   .Cells(1, 1).Value = "Stock Group"
   Row_Num = 3
   For StockGroup = 1 To NumSGroups
      .Cells(Row_Num, 1).Value = StockGroupNames(StockGroup) 'stock names start in 2nd row
      Row_Num = Row_Num + 1
   Next StockGroup
   .Cells(Row_Num, 1).Value = "Total"
   'Fit the column width to the headings
   .Columns(1).AutoFit
End With
Catch_WkBk.Worksheets.Add after:=Worksheets(3)
Catch_WkBk.Worksheets(4).Name = "FSCCatchByStock"
With Catch_WkBk.Worksheets("FSCCatchByStock")
   .Cells(1, 1).Value = "Stock"
   Row_Num = 3
   For Stock = 1 To NumStocks
      .Cells(Row_Num, 1).Value = StockNames(Stock) 'stock names start in 2nd row
      Row_Num = Row_Num + 1
   Next Stock
   .Cells(Row_Num, 1).Value = "Total"
   'Fit the column width to the headings
   .Columns(1).AutoFit
End With
Catch_WkBk.Worksheets.Add after:=Worksheets(4)
Catch_WkBk.Worksheets(5).Name = "ESSRCatchByStock"
With Catch_WkBk.Worksheets("ESSRCatchByStock")
   .Cells(1, 1).Value = "Stock"
   Row_Num = 3
   For Stock = 1 To NumStocks
      .Cells(Row_Num, 1).Value = StockNames(Stock) 'stock names start in 2nd row
      Row_Num = Row_Num + 1
   Next Stock
   .Cells(Row_Num, 1).Value = "Total"
   'Fit the column width to the headings
   .Columns(1).AutoFit
End With
'terminal escapement size workbook
testfile = Dir("Escape.xls")
If Len(testfile) > 0 Then Kill ("Escape.xls")
Set Escape_WkBk = Workbooks.Add
Escape_WkBk.SaveAs Filename:="Escape.xls", FileFormat:=xlAddIn   'Later => prompt for output filename
'Initialize stock group escapement size worksheet
Escape_WkBk.Worksheets(1).Name = "StockGroup"
With Escape_WkBk.Worksheets("StockGroup")
   .Cells(1, 1).Value = "Stock Group"
```

```
    Row_Num = 3
    For StockGroup = 1 To NumSGroups
        .Cells(Row_Num, 1).Value = StockGroupNames(StockGroup) 'stock names start in 3'rd row
        Row_Num = Row_Num + 1
    Next StockGroup
    .Cells(Row_Num, 1).Value = "Total"
    'Fit the column width to the headings
    .Columns(1).AutoFit
End With
'Stock Calc Entry run size workbook  (CNoble Oct 11)
testfile = Dir("Stock_Calc_Entry.xls")
If Len(testfile) > 0 Then Kill ("Stock_Calc_Entry.xls")
Set Stock_Calc_Entry_WkBk = Workbooks.Add
Stock_Calc_Entry_WkBk.SaveAs Filename:="Stock_Calc_Entry.xls", FileFormat:=xlAddIn   'Later => prompt for
output filename
'Initialize stock group escapement size worksheet
Stock_Calc_Entry_WkBk.Worksheets(1).Name = "Stock"
With Stock_Calc_Entry_WkBk.Worksheets("Stock")
    .Cells(1, 1).Value = "Stock"
    Row_Num = 3
    For Stock = 1 To NumStocks
        .Cells(Row_Num, 1).Value = StockNames(Stock) 'stock names start in 3'rd row
        Row_Num = Row_Num + 1
    Next Stock
    '.Cells(Row_Num, 1).Value = "Total"
    'Fit the column width to the headings
    .Columns(1).AutoFit
End With
'Initialize stock escapement size worksheet
Escape_WkBk.Worksheets(2).Name = "Stock"
With Escape_WkBk.Worksheets("Stock")
    .Cells(1, 1).Value = "Stock"
    Row_Num = 3
    For Stock = 1 To NumStocks
        .Cells(Row_Num, 1).Value = StockNames(Stock) 'stock names start in 3'rd column
        Row_Num = Row_Num + 1
    Next Stock
    .Cells(Row_Num, 1).Value = "Total"
    'Fit the column width to the headings
    .Columns(1).AutoFit
End With
'Exploitation rate workbook
testfile = Dir("ExploitRate.xls")
If Len(testfile) > 0 Then Kill ("ExploitRate.xls")
Set ExploitRate_WkWb = Workbooks.Add
ExploitRate_WkWb.SaveAs Filename:="ExploitRate.xls", FileFormat:=xlAddIn   'Later => prompt for output
filename
'Initialize stock group harvest rate worksheet
ExploitRate_WkWb.Worksheets(1).Name = "StockGroup"
With ExploitRate_WkWb.Worksheets("StockGroup")
    .Cells(1, 1).Value = "Stock Group"
    Row_Num = 3
    For StockGroup = 1 To NumSGroups
        .Cells(Row_Num, 1).Value = StockGroupNames(StockGroup) 'stock group names start in 3nd row
        Row_Num = Row_Num + 1
    Next StockGroup
    .Cells(Row_Num, 1).Value = "Total"
    'Fit the column width to the headings
```

```
    .Columns(1).AutoFit
    'Format all cells as text to force display of decimal place when 0
    .Range(.Cells(3, 2), .Cells(NumSGroups + 3, EndYear - StartYear + Year_Datacol)).Columns.NumberFormat =
"0.0"
End With
'Initialize stock exploitation rate worksheet
ExploitRate_WkWb.Worksheets(2).Name = "Stock"
With ExploitRate_WkWb.Worksheets("Stock")
    .Cells(1, 1).Value = "Stock"
    Row_Num = 3
    For Stock = 1 To NumStocks
        .Cells(Row_Num, 1).Value = StockNames(Stock) 'stock names start in 3nd row
        Row_Num = Row_Num + 1
    Next Stock
    .Cells(Row_Num, 1).Value = "Total"
    'Fit the column width to the headings
    .Columns(1).AutoFit
    'Format all cells as text to force display of decimal place when 0
    .Range(.Cells(3, 2), .Cells(NumStocks + 3, EndYear - StartYear + Year_Datacol)).Columns.NumberFormat =
"0.0"
End With
'daily fishery harvest rate workbook
testfile = Dir("FisheryHr.xls")
If Len(testfile) > 0 Then Kill ("FisheryHr.xls")
Set FisheryHr_WkBk = Workbooks.Add
FisheryHr_WkBk.SaveAs Filename:="FisheryHR.xls", FileFormat:=xlAddIn   'Later => prompt for output filename
With FisheryHr_WkBk
    Sheet = 1
    SheetCount = .Worksheets.Count
    For Year = StartYear To EndYear
        'Check if need to add a worksheet to hold this year's reconstruction output
        If Sheet > SheetCount Then
            .Worksheets.Add after:=Worksheets(Sheet - 1)
        End If
        .Worksheets(Sheet).Name = Right(Str(Year), 4) 'str function adds leading blank - remove by extract 4
rightmost characters
        'Initialize the column headings
        With .Worksheets(Sheet)
            .Cells(1, 1).Value = "Week"
            For iweek = 1 To WeeksInModel
                .Cells(FishHR_Datarow + iweek - 1, 1).Value = iweek 'week numbers are in first column
            Next iweek
            For Fishery = 1 To NumFisheries
                .Cells(1, FishHR_FishCol + Fishery - 1).Value = FisheryNames(Fishery) 'fishery names start in 2nd
column
            Next Fishery
            .Range(.Cells(1, 1), .Cells(1, NumFisheries + 1)).ColumnWidth = 6
            .Range(.Cells(1, 1), .Cells(1, NumFisheries + 1)).Orientation = 90
        End With
        Sheet = Sheet + 1
    Next Year
End With
'weekly harvest rate workbook
testfile = Dir("WeeklyHr.xls")
If Len(testfile) > 0 Then Kill ("WeeklyHr.xls")
Set WeeklyHr_WkBk = Workbooks.Add
WeeklyHr_WkBk.SaveAs Filename:="WeeklyHR.xls", FileFormat:=xlAddIn   'Later => prompt for output filename
With WeeklyHr_WkBk
```

```
    Sheet = 1
    SheetCount = .Worksheets.Count
    For Year = StartYear To EndYear
        'Check if need to add a worksheet to hold this year's reconstruction output
        If Sheet > SheetCount Then
            .Worksheets.Add after:=Worksheets(Sheet - 1)
        End If
        .Worksheets(Sheet).Name = Right(Str(Year), 4) 'str function adds leading blank - remove by extract 4
rightmost characters
        'Initialize the column headings
        With .Worksheets(Sheet)
            .Cells(1, 1).Value = "Week"
            For iweek = 1 To WeeksInModel
                .Cells(FishHR_Datarow + iweek - 1, 1).Value = iweek 'week numbers are in first column
            Next iweek
            For Fishery = 1 To NumFisheries
                .Cells(1, FishHR_FishCol + Fishery - 1).Value = FisheryNames(Fishery) 'fishery names start in 2nd
column
            Next Fishery
            .Range(.Cells(1, 1), .Cells(1, NumFisheries + 1)).ColumnWidth = 6
            .Range(.Cells(1, 1), .Cells(1, NumFisheries + 1)).Orientation = 90
        End With
        Sheet = Sheet + 1
    Next Year
End With
   'weekly unexplained catch
testfile = Dir("Unexplained.xls")
If Len(testfile) > 0 Then Kill ("Unexplained.xls")
Set Unexpl_WkBk = Workbooks.Add
Unexpl_WkBk.SaveAs Filename:="Unexplained.xls", FileFormat:=xlAddIn   'Later => prompt for output filename
With Unexpl_WkBk
    Sheet = 1
    SheetCount = .Worksheets.Count
    For Year = StartYear To EndYear
        'Check if need to add a worksheet to hold this year's reconstruction output
        If Sheet > SheetCount Then
            .Worksheets.Add after:=Worksheets(Sheet - 1)
        End If
        .Worksheets(Sheet).Name = Right(Str(Year), 4) 'str function adds leading blank - remove by extract 4
rightmost characters
        'Initialize the column headings
        With .Worksheets(Sheet)
            .Cells(1, 1).Value = "Week"
            For Week = 1 To WeeksInModel
                .Cells(FishHR_Datarow + Week - 1, 1).Value = Week 'week numbers are in first column
            Next Week
            .Cells(FishHR_Datarow + WeeksInModel, 1).Value = "Total"
            For Fishery = 1 To NumFisheries
                .Cells(1, FishHR_FishCol + Fishery - 1).Value = FisheryNames(Fishery) 'fishery names start in 2nd
column
            Next Fishery
            .Range(.Cells(1, 1), .Cells(1, NumFisheries + 1)).ColumnWidth = 6
            .Range(.Cells(1, 1), .Cells(1, NumFisheries + 1)).Orientation = 90
        End With
        Sheet = Sheet + 1
    Next Year
End With
End Sub
```

```
Sub FishResSpawn(Yr)
'Reads in the fishery residence times for each stock and determines the cumulative time from tributary escapment to
'each fishery.
'Reads in Tyee peak, stock offset and stock SD for the year.
'Reads in the entry size by stock for the year
'data offsets in fish residence times worksheet
Const fishres_datarow As Integer = 3
Const stockgrp_col As Integer = 4
Const fishery1_col As Integer = 7
'data offsets in fish spawning timing worksheet
Const Timing_off_col As Integer = 5     'Column number for offset
Const Timing_off_row As Integer = 5     'Row number for offset
Const Timing_med_col As Integer = 2    'Column number for Tyee median
Const Timing_med_row As Integer = 2    'row for Base Year
'data offsets in entry size
Const Entry_row As Integer = 6
Const Entry_col As Integer = 2
Dim cum, Fishery, fishres_row, ResTime, Stock, StockGroup As Integer 'local variables for residence data
Dim timing_row, trib_time As Integer 'local variables for spawning timing data
Dim Peak, date1, date2, x
'Read in the fishery residence times for each stock.
Sht_res = "FISHRES"
fishres_row = fishres_datarow 'Skip the header rows
With Timing_WkBk.Worksheets(Sht_res)
   For Stock = 1 To NumStocks
      StockToStockGroup(Stock) = .Cells(fishres_row, stockgrp_col).Value
      For Fishery = 1 To NumFisheries
         'Test for empty cell => causes crash
         If IsEmpty(.Cells(fishres_row, fishery1_col + Fishery - 1).Value) Or _
            (.Cells(fishres_row, fishery1_col + Fishery - 1).Value = "") Then
            ResTime = 0
         Else
            ResTime = .Cells(fishres_row, fishery1_col + Fishery - 1).Value
         End If
         Residence(Stock, Fishery) = ResTime
      Next Fishery
      fishres_row = fishres_row + 1
   Next Stock
End With
'Calculate the cumulative residence times from Tyee entry for each stock and fishery
For Stock = 1 To NumStocks
   cum = 0
   For Fishery = 1 To NumFisheries
      cum = cum + Residence(Stock, Fishery)
      Cum_ResTimes(Stock, Fishery) = cum
   Next Fishery
Next Stock
'read in Tyee peak
date1 = Timing_WkBk.Worksheets("TIMING_Mean").Cells(Timing_med_row + Yr - BaseYear, Timing_med_col)
date2 = DateValue("June 1, " & Format(Year(date1)))
Peak = DateDiff("d", date2, date1) + 1
'Read in the SD and offset.
Sht_res = "TIMING"
With Timing_WkBk.Worksheets(Sht_res)
   For Stock = 1 To NumStocks
      Spawn_Peak(Stock) = .Cells(Timing_off_row + Stock - 1, Timing_off_col)
      Spawn_Peak(Stock) = Spawn_Peak(Stock) + Peak
      Spawn_SD(Stock) = .Cells(Timing_off_row + Stock - 1, Timing_off_col + 2 + Yr - BaseYear)
```

```
    Next Stock
End With
'Read in the entry size by stock
Sht_res = "Skeena Escape"
With Entry_WkBk.Worksheets(Sht_res)
    For Stock = 1 To NumStocks
        x = .Cells(Entry_row + Yr - BaseYear, Entry_col + Stock - 1).Value
        If IsEmpty(x) Or x = "" Then
            Spawn_Entry(Stock) = 0
        Else
            Spawn_Entry(Stock) = x
        End If
    Next Stock
End With
'Add terminal catch and escapement to get entry stock size for Sustut and Bulkley sockeye (KKE - 22 Jan. 2013)
'Spawn Entry numbers for Babine stocks include catches at and above the Babine fence
        Spawn_Entry(3) = Spawn_Entry(3) + Total_Catch(12)          'Sustut
        Spawn_Entry(14) = Spawn_Entry(14) + Total_Catch(11)        'Bulkley
End Sub
Sub Calc_Escape(Yr)
'Calculates the daily escapement for each stock in the year specified
Dim curdate As Date, bdate As Date
Dim irow As Integer, iday As Integer, Stock As Integer, Sector As Integer, isg As Integer
Dim agg As Double, psum As Double
Dim p(NumStocks) As Double
Const Tyee_row As Integer = 5
Const Tyee_col As Integer = 1
irow = Tyee_row
Erase Stock_Calc_Entry                    ' CNoble Oct 11
With Fisheries_WkBk.Worksheets("TyeeRunbyDay")
    curdate = .Cells(irow, Tyee_col)
    Do While curdate > 0
        bdate = DateValue("June 1," & Format(Year(curdate)))
        iday = DateDiff("d", bdate, curdate) + 1
        agg = .Cells(irow, Tyee_col + Yr - BaseYear + 1)
        psum = 0
        For Stock = 1 To NumStocks
            p(Stock) = Spawn_Entry(Stock) * WorksheetFunction.NormDist(iday, Spawn_Peak(Stock),
Spawn_SD(Stock), False)
            psum = psum + p(Stock)
        Next Stock
        For Stock = 1 To NumStocks
            If psum > 0 Then
                Stock_Daily_Escape(Stock, iday) = p(Stock) * agg / psum
                Stock_Calc_Entry(Stock) = Stock_Calc_Entry(Stock) + Stock_Daily_Escape(Stock, iday)
            Else
                Stock_Daily_Escape(Stock, iday) = 0
            End If
        Next Stock
        irow = irow + 1
        curdate = .Cells(irow, Tyee_col)
    Loop
End With
'totals
Erase StockGroupTrun, StockTrun
RunTotal = 0
For Stock = 1 To NumStocks
    For iday = 1 To DaysInModel
```

```
      agg = Stock_Daily_Escape(Stock, iday)
      StockTrun(Stock) = StockTrun(Stock) + agg
      isg = StockToStockGroup(Stock)
      StockGroupTrun(isg) = StockGroupTrun(isg) + agg
   Next iday
   RunTotal = RunTotal + StockTrun(Stock)
Next Stock
Erase CatchBySector                    ' CNoble Oct 11
End Sub
Function gfs(HRate, T, Fshry, FDaysPerWk) As Double
'Returns catch for the given harvest rate (forward algorithm with entry timing)
Dim Day, i, k, ResidanceDays, Stock As Integer
Dim CaTest, x, surv(50) As Double
CaTest = 0
For Stock = 1 To NumStocks
   'Only do if the stock go through the fishery, i.e. residence time > 0
   ResidanceDays = Residence(Stock, Fshry)
   If ResidanceDays > 0 Then
      'Initialize survival to 100 percent
      For i = 1 To FDaysPerWk + ResidanceDays + 1
         surv(i) = 1
      Next i
      'Work forward through the fishing days
      For Day = 1 To FDaysPerWk
         For k = 1 To ResidanceDays
            'i is fishery chunks being harvested
            'j is escapement day
            i = Day + k - 2
            j = i + T - Cum_ResTimes(Stock, Fshry)
            'Pick up chunk
            If (j < 1 Or j > DaysInModel) Then
               x = 0
            Else
               x = Stock_Daily_Escape(Stock, j)
            End If
            'Calculate survival and catch
            If (HRate < 1) And (x > 0) Then
               CaTest = CaTest + HRate * surv(i + 1) * x
               surv(i + 1) = surv(i + 1) * (1 - HRate)
            End If
         Next k
      Next Day
   End If
Next Stock
gfs = CaTest
End Function
Function CalcHarvestRate(Wk, Yr, Fishry, FDaysPerWeek, Catch, TeeDay) As Double
Const Tol As Double = 0.001
Dim j, k, Stock As Integer
Dim ESum, HarvestRate, High, Low, test As Double
'Calculates the harvest rate for a fishery
HarvestRate = 0
'If only one day of fishing, sum up the escapement
If FDaysPerWeek = 1 Then
   ESum = 0
   For Stock = 1 To NumStocks
      For k = 1 To Residence(Stock, Fishry)
         j = TeeDay + k - 1 - Cum_ResTimes(Stock, Fishry)
```

```
            If (j < 1 Or j < DaysInModel) Then
               ESum = ESum + Stock_Daily_Escape(Stock, j)
            End If
         Next k
      Next Stock
      If (ESum = 0) Then
         HarvestRate = -0.99
      Else
         HarvestRate = Catch / (Catch + ESum)
      End If
   Else
      'Use the bisection method to estimate the harvest rate
      Low = 0
      High = 1
      HarvestRate = 0.5 'Start with 50% harvest
      test = 0
      Do Until (Abs(Catch - test) / Catch * 100) < Tol
         test = gfs(HarvestRate, TeeDay, Fishry, FDaysPerWeek)
         If test = 0 Then
            HarvestRate = -0.99
            Exit Do
         End If
         If Catch > test Then
            Low = HarvestRate
         Else
            High = HarvestRate
         End If
         HarvestRate = (Low + High) / 2
      Loop
   End If
   CalcHarvestRate = HarvestRate
   End Function
   Sub Reconstruct(Yr, Year_String)
   'Does a forward reconstruction for the given year.
   Dim Day As Integer, EscDay As Integer, Fishery As Integer, FishDays As Integer
   Dim ResDay As Integer, StockGroup As Integer, Week As Integer
   Dim interc(NumFisheries, NumStocks) As Double  'Total catch for each stock in each fishery
   Dim HarvestRate As Double, FisheryWkHR As Double, StockCatch As Double, PSport As Double
   Dim WklyCat(WeeksInModel) As Double, WklyAA(WeeksInModel) As Double, WklyUnexpl(WeeksInModel) As
   Double
   Dim iweek As Integer
   Dim WeeklyCatch As Double, pFSC As Double, TerminalRun As Double
   OutputRow = 2  'Initialize row to begin writing data in the run reconstruction workbook
   Erase interc
   Erase TerminalHR
   savcat = 0: predcat = 0
   For Fishery = 1 To NumFisheries
      Erase WklyCat, WklyAA, WklyUnexpl
      FishDays = 7    'fishing 7 days a week
      'Calcululate harvest rate for each terminal fishery (KKE - 14 Oct. 2012)
      If Fishery >= 9 And Fishery <= 12 Then
         If Fishery = 9 Then TerminalRun = Spawn_Entry(10)
         If Fishery = 10 Then TerminalRun = Spawn_Entry(11)
         If Fishery = 11 Then TerminalRun = Spawn_Entry(13) + Spawn_Entry(14)
         If Fishery = 12 Then TerminalRun = Spawn_Entry(3)
         TerminalHR(Fishery) = 0
         If TerminalRun > 0 Then TerminalHR(Fishery) = Total_Catch(Fishery) / TerminalRun
      End If
```

```
'Work forwards through each week of the fishery
Erase FisheryHR
For iweek = 1 To WeeksInModel
   WeeklyCatch = FSC_Catch(Fishery, iweek) + ESSR_Catch(Fishery, iweek)
   If WeeklyCatch > 0 Then
      pFSC = FSC_Catch(Fishery, iweek) / WeeklyCatch
      savcat = savcat + WeeklyCatch
      'Calculate the harvest rate for todays catch in this fishery
      Tday = Week_CatchDay(iweek)
      'Use terminal harvest rate for Pinkut, Fulton, Bulkley-Morice and Sustut Fisheries (KKE 14 Oct. 2012)
      If Fishery >= 9 And Fishery <= 12 Then
        HarvestRate = TerminalHR(Fishery)
      Else
        HarvestRate = CalcHarvestRate(iweek, Yr, Fishery, FishDays, WeeklyCatch, Tday)
      End If
      'Save Daily Harvest Rate - 15 March 2012
      FisheryHR(Fishery, iweek) = HarvestRate
      'Update Run Size
      If HarvestRate > 0 Then
         For Stock = 1 To NumStocks
            For ResDay = 1 To Residence(Stock, Fishery)
               For Day = 1 To FishDays
                  EscDay = Tday - Cum_ResTimes(Stock, Fishery) + ResDay + Day - 2
                  If (EscDay > 0 And EscDay <= DaysInModel) Then
                     'Calculate the catch for this stock and update the yearly catch for it
                     StockCatch = HarvestRate * Stock_Daily_Escape(Stock, EscDay)
                     StockYearCatch(Stock) = StockYearCatch(Stock) + StockCatch
                     WklyCat(iweek) = WklyCat(iweek) + StockCatch
                     'overwrite the stock daily escapement with the number of fish aftet harvesting
                     Stock_Daily_Escape(Stock, EscDay) = Stock_Daily_Escape(Stock, EscDay) * (1 - HarvestRate)
                     'Update the available abundance into this fishery for this week
                     WklyAA(iweek) = WklyAA(iweek) + Stock_Daily_Escape(Stock, EscDay)
                     'Sum catch by sector
                     CatchBySector(Stock, 1) = CatchBySector(Stock, 1) + StockCatch * pFSC
                     CatchBySector(Stock, 2) = CatchBySector(Stock, 2) + StockCatch * (1 - pFSC)
                     'update the catch for this stock group and fishery
                     StockGroup = StockToStockGroup(Stock)
                     interc(Fishery, Stock) = interc(Fishery, Stock) + StockCatch
                     predcat = predcat + StockCatch
                  End If
               Next Day
            Next ResDay
         Next Stock
      Else
         If HarvestRate < 0 Then WklyUnexpl(iweek) = WklyUnexpl(iweek) + WeeklyCatch
      End If
   End If
 Next iweek
'write to the daily fisheries harvest rate workbook
For iweek = 1 To WeeksInModel
   hr = FisheryHR(Fishery, iweek) * 100
   FisheryHr_WkBk.Worksheets(Year_String).Cells(FishHR_Datarow + iweek - 1, FishHR_FishCol + Fishery -
1).Value = Format(hr, "##0.0") 'Round to 1 decimal place
Next iweek

'Calculate the weekly harvest rate for this fishery and write to the fisheries harvest rate workbook
UnexplSum = 0
For iweek = 1 To WeeksInModel
```

```
      aa = WklyAA(iweek)
      cc = WklyCat(iweek)
      If aa > 0 Then
         FisheryWkHR = cc / (cc + aa) * 100
      Else
         FisheryWkHR = 0  ' Do we want to print this or just leave blank?
      End If
      WeeklyHr_WkBk.Worksheets(Year_String).Cells(FishHR_Datarow + iweek - 1, FishHR_FishCol + Fishery -
1).Value = Format(FisheryWkHR, "##0.0") 'Round to 1 decimal place
      If WklyUnexpl(iweek) > 0 Then Unexpl_WkBk.Worksheets(Year_String).Cells(FishHR_Datarow + iweek - 1,
FishHR_FishCol + Fishery - 1).Value = Round(WklyUnexpl(iweek))
      UnexplSum = UnexplSum + WklyUnexpl(iweek)
   Next iweek
   If UnexplSum > 0 Then Unexpl_WkBk.Worksheets(Year_String).Cells(FishHR_Datarow + WeeksInModel,
FishHR_FishCol + Fishery - 1).Value = Round(UnexplSum)
    'Output the catch for each stock in this fishery to the run reconstruction workbook
    With Reconstruct_WkBk.Worksheets(Year_String)
       .Cells(OutputRow, 1).Value = FisheryNames(Fishery)
       CatchTotal = 0
       For Stock = 1 To NumStocks
          StockCatch = interc(Fishery, Stock)
          CatchTotal = CatchTotal + StockCatch
          StockGroup = StockToStockGroup(Stock)
          StockGroupYearCatch(StockGroup) = StockGroupYearCatch(StockGroup) + StockCatch    'Sum the catch
for all fisheries for this stock group
          .Cells(OutputRow, Stock + 1).Value = Round(StockCatch) 'Round to nearest integer for output
       Next Stock
       .Cells(OutputRow, Stock + 1).Value = Round(CatchTotal) 'Output total catch for this fishery
    End With
    OutputRow = OutputRow + 1
Next Fishery
End Sub
Sub OutputData(Yr, Year_String, YearCol)
Dim Day As Integer, Stock As Integer, StockGroup As Integer, StockCatch As Long
Dim StockEsc(NumStocks) As Double, StockGroupEsc(NumSGroups) As Double   'Total ending escapement for
each stock and stock group
Dim TotalSectorCatch(NumSectors) As Double                'Total Catch by Sector
Dim GroupCatchBySector(NumSGroups, NumSectors) As Double       'Total Catch by Group and Sector
Dim HR_Total As Double, Stock_HR As Double
Dim CatchTotal As Double, SCatchTotal As Double, TotalEscape As Double
Dim harvest_rate As String
Erase StockEsc, StockGroupEsc
TotalEscape = 0
For Stock = 1 To NumStocks
    StockGroup = StockToStockGroup(Stock)
    For Day = 1 To DaysInModel
       xxx = Stock_Daily_Escape(Stock, Day)
       StockEsc(Stock) = StockEsc(Stock) + xxx
       StockGroupEsc(StockGroup) = StockGroupEsc(StockGroup) + xxx
    Next Day
    TotalEscape = TotalEscape + StockEsc(Stock)
Next Stock
'Write the Stock Calc Entry data (CNoble Oct 11)
'Reconstruct_WkBk.Worksheets(Year_String).Cells(Reconst_CatchRow, 1).Value = "Catch"
Stock_Calc_Entry_WkBk.Worksheets("Stock").Cells(1, YearCol).Value = Year_String
For Stock = 1 To NumStocks
    Stock_Calc_Entry(Stock) = Round(Stock_Calc_Entry(Stock)) 'Round to nearest integer
    'Reconstruct_WkBk.Worksheets(Year_String).Cells(Reconst_CatchRow, StockGroup + 1).Value = StockCatch
```

```
    Stock_Calc_Entry_WkBk.Worksheets("Stock").Cells(Stock + 2, YearCol).Value = Stock_Calc_Entry(Stock)
Next Stock
'Write the total catch for each stock group to both the Run Reconstruction and Annual Terminal Catch workbooks
CatchTotal = 0
Reconstruct_WkBk.Worksheets(Year_String).Cells(Reconst_CatchRow, 1).Value = "Catch"
Catch_WkBk.Worksheets("Catch").Cells(1, YearCol).Value = Year_String
For StockGroup = 1 To NumSGroups
    StockCatch = Round(StockGroupYearCatch(StockGroup)) 'Round to nearest integer
    Catch_WkBk.Worksheets("Catch").Cells(StockGroup + 2, YearCol).Value = StockCatch
    CatchTotal = CatchTotal + StockCatch
Next StockGroup
For Stock = 1 To NumStocks
    Reconstruct_WkBk.Worksheets(Year_String).Cells(Reconst_CatchRow, Stock + 1).Value =
Round(StockYearCatch(Stock))
Next Stock
'Write out the total catch across all stock groups
Reconstruct_WkBk.Worksheets(Year_String).Cells(Reconst_CatchRow, NumStocks + 2).Value = CatchTotal
Catch_WkBk.Worksheets("Catch").Cells(NumSGroups + 3, YearCol).Value = CatchTotal
SCatchTotal = CatchTotal
'Write the catch by sector for each stock group to the annual catch workbook
For Stock = 1 To NumStocks
    StockGroup = StockToStockGroup(Stock)
    For Sector = 1 To NumSectors
        GroupCatchBySector(StockGroup, Sector) = GroupCatchBySector(StockGroup, Sector) +
CatchBySector(Stock, Sector)
    Next Sector
Next Stock
CatchTotal = 0
Catch_WkBk.Worksheets("FSCCatchByGroup").Cells(1, YearCol).Value = Year_String
For StockGroup = 1 To NumSGroups
    StockCatch = Round(GroupCatchBySector(StockGroup, 1))  'Round to nearest integer
    Catch_WkBk.Worksheets("FSCCatchByGroup").Cells(StockGroup + 2, YearCol).Value = StockCatch
    CatchTotal = CatchTotal + StockCatch
Next StockGroup
'Write out the total catch across all stock groups
Catch_WkBk.Worksheets("FSCCatchByGroup").Cells(NumSGroups + 3, YearCol).Value = CatchTotal
CatchTotal = 0
Catch_WkBk.Worksheets("ESSRCatchByGroup").Cells(1, YearCol).Value = Year_String
For StockGroup = 1 To NumSGroups
    StockCatch = Round(GroupCatchBySector(StockGroup, 2))  'Round to nearest integer
    Catch_WkBk.Worksheets("ESSRCatchByGroup").Cells(StockGroup + 2, YearCol).Value = StockCatch
    CatchTotal = CatchTotal + StockCatch
Next StockGroup
'Write out the total catch across all stock groups
Catch_WkBk.Worksheets("ESSRCatchByGroup").Cells(NumSGroups + 3, YearCol).Value = CatchTotal
'Calculate the total run and harvest rates for each stock group and write to the Run Reconstruction,
'Annual Terminal Run and Harvest Rate workbooks
Reconstruct_WkBk.Worksheets(Year_String).Cells(Reconst_CatchRow + 1, 1).Value = "Run"
Reconstruct_WkBk.Worksheets(Year_String).Cells(Reconst_CatchRow + 2, 1).Value = "ER"
Reconstruct_WkBk.Worksheets(Year_String).Rows(Reconst_CatchRow + 2).NumberFormat = "0.0" 'Format cells
to force display of decimal place when 0
Reconstruct_WkBk.Worksheets(Year_String).Columns(1).AutoFit
ExploitRate_WkWb.Worksheets("StockGroup").Cells(1, YearCol).Value = Year_String
ExploitRate_WkWb.Worksheets("Stock").Cells(1, YearCol).Value = Year_String
Escape_WkBk.Worksheets("Stockgroup").Cells(1, YearCol).Value = Year_String
Escape_WkBk.Worksheets("Stock").Cells(1, YearCol).Value = Year_String
Catch_WkBk.Worksheets("FSCCatchByGroup").Cells(1, YearCol).Value = Year_String
Catch_WkBk.Worksheets("ESSRCatchByGroup").Cells(1, YearCol).Value = Year_String
```

```
Catch_WkBk.Worksheets("FSCCatchByStock").Cells(1, YearCol).Value = Year_String
Catch_WkBk.Worksheets("ESSRCatchByStock").Cells(1, YearCol).Value = Year_String
For Sector = 1 To NumSectors
   TotalSectorCatch(Sector) = 0
Next Sector
For Stock = 1 To NumStocks
   Catch_WkBk.Worksheets("FSCCatchByStock").Cells(Stock + 2, YearCol).Value = Round(CatchBySector(Stock,
1))
   Catch_WkBk.Worksheets("ESSRCatchByStock").Cells(Stock + 2, YearCol).Value =
Round(CatchBySector(Stock, 2))
   For Sector = 1 To NumSectors
      TotalSectorCatch(Sector) = TotalSectorCatch(Sector) + Round(CatchBySector(Stock, Sector))
   Next Sector
Next Stock
Catch_WkBk.Worksheets("FSCCatchByStock").Cells(Stock + 2, YearCol).Value = TotalSectorCatch(1)
Catch_WkBk.Worksheets("ESSRCatchByStock").Cells(Stock + 2, YearCol).Value = TotalSectorCatch(2)
For StockGroup = 1 To NumSGroups
   'Calculate the harvest rate for this stock group
   If StockGroupTrun(StockGroup) > 0 Then
      Stock_HR = (StockGroupYearCatch(StockGroup) / StockGroupTrun(StockGroup)) * 100
   Else
      Stock_HR = 0
   End If
   StockGroupYearCatch(StockGroup) = 0   'Zero for the next year calculation
   harvest_rate = Format(Stock_HR, "###.0") 'Round to 1 decimal place
   'Reconstruct_WkBk.Worksheets(Year_String).Cells(Reconst_CatchRow + 2, StockGroup + 1).Value =
harvest_rate
   ExploitRate_WkWb.Worksheets("StockGroup").Cells(StockGroup + 2, YearCol).Value = Stock_HR
   Escape_WkBk.Worksheets("StockGroup").Cells(StockGroup + 2, YearCol).Value =
Round(StockGroupEsc(StockGroup))
Next StockGroup
For Stock = 1 To NumStocks
   Reconstruct_WkBk.Worksheets(Year_String).Cells(Reconst_CatchRow + 1, Stock + 1).Value =
Round(StockTrun(Stock))
   If StockTrun(Stock) > 0 Then
      Stock_HR = StockYearCatch(Stock) / StockTrun(Stock) * 100
   Else
      Stock_HR = 0
   End If
   Reconstruct_WkBk.Worksheets(Year_String).Cells(Reconst_CatchRow + 2, Stock + 1).Value =
Format(Stock_HR, "###.0")
Next Stock
'Calculate the harvest rate for each stock and write to the stock harvest rate worksheet
For Stock = 1 To NumStocks
   If StockTrun(Stock) > 0 Then
      Stock_HR = (StockYearCatch(Stock) / StockTrun(Stock)) * 100
   Else
      Stock_HR = 0
   End If
   StockYearCatch(Stock) = 0   'Zero for the next year calculation
   For Sector = 1 To NumSectors
      CatchBySector(Stock, Sector) = 0
   Next Sector
   ExploitRate_WkWb.Worksheets("Stock").Cells(Stock + 2, YearCol).Value = Stock_HR
   Escape_WkBk.Worksheets("Stock").Cells(Stock + 2, YearCol).Value = Round(StockEsc(Stock))
Next Stock
'Write out the total run across all stock groups
```

```
Reconstruct_WkBk.Worksheets(Year_String).Cells(Reconst_CatchRow + 1, NumStocks + 2).Value =
Round(RunTotal)
'Write out the total harvest rate across all stock groups
If RunTotal > 0 Then
   HR_Total = SCatchTotal / RunTotal * 100
Else
   HR_Total = 0
End If
harvest_rate = Format(HR_Total, "###.0") 'Round to 1 decimal place
Reconstruct_WkBk.Worksheets(Year_String).Cells(Reconst_CatchRow + 2, NumStocks + 2).Value = harvest_rate
ExploitRate_WkWb.Worksheets("StockGroup").Cells(NumSGroups + 3, YearCol).Value = HR_Total
ExploitRate_WkWb.Worksheets("Stock").Cells(NumStocks + 3, YearCol).Value = HR_Total
Escape_WkBk.Worksheets("StockGroup").Cells(NumSGroups + 3, YearCol).Value = Round(TotalEscape)
Escape_WkBk.Worksheets("Stock").Cells(NumStocks + 3, YearCol).Value = Round(TotalEscape)
End Sub
```