Statistical Analysis of 2006 Lower Granite Dam Fall Chinook Run Reconstruction

Report for PSC Southern Boundary Restoration and Enhancement Fund Project:

Lower Granite Fall Chinook Run Reconstruction Assistance (Phase 2)

Kirk Steinhorst
Department of Statistics
University of Idaho
Moscow, ID 83844-1104
kirk@uidaho.edu

Deborah Milks
Washington Department of Fish and Wildlife
Snake River Lab
401 S. Cottonwood
Dayton, WA 99328
milksdjm@dfw.wa.gov

George Naughton
Department of Fish and Wildlife
University of Idaho
Moscow, ID 83844
naughton@uidaho.edu

and

Bill Arnsberg
Nez Perce Tribe
P.O. Box 365
Lapwai, ID 83540
billa@nezperce.org

August 11, 2007

I. Background

In Phase 1 of this study, we developed bootstrap confidence intervals for groups, including wild fish, of fall Chinook salmon, *Oncorhynchus tshawytscha*, returning to Lower Granite dam near Lewiston, ID and Clarkston, WA. This statistical analysis depended on 3 data sets—daily window counts at the dam, data collected from fish trapped at the dam, and data from fish processed at Lyons Ferry Hatchery or the Nez Perce Tribal Hatchery. The result was 90% confidence intervals for each group.

The window count data were used to estimate the numbers of adults and jacks arriving at the dam before and after the trap at the dam was operated. In this case, window counts supplied numbers from August 18th to September 5 and from November 21st to December 15th. The trap was run from September 6th to November 20th. Since the trap was open 13% of the time, dividing the numbers of fish trapped by 13% provides an estimate of fish arriving at the dam for the trapping period. Data collected on trapped fish include sex, length, markings, presence or absence of coded wire tags and PIT tags, and approximately 66% of the untagged fish were scale sampled for age and origin determinations. Some trapped fish were released at the dam and the rest were transported to the hatcheries for processing. More observations were made at the hatcheries of sex, length, and markings and, in addition, coded wire tags were retrieved and read and PIT tags were read and recorded. Scales were collected on all untagged fish. The processing data were used to assign group origin (including wild) to each fish. Fish released at the dam during the trapping period were assigned origins from scale samples taken at the trap.

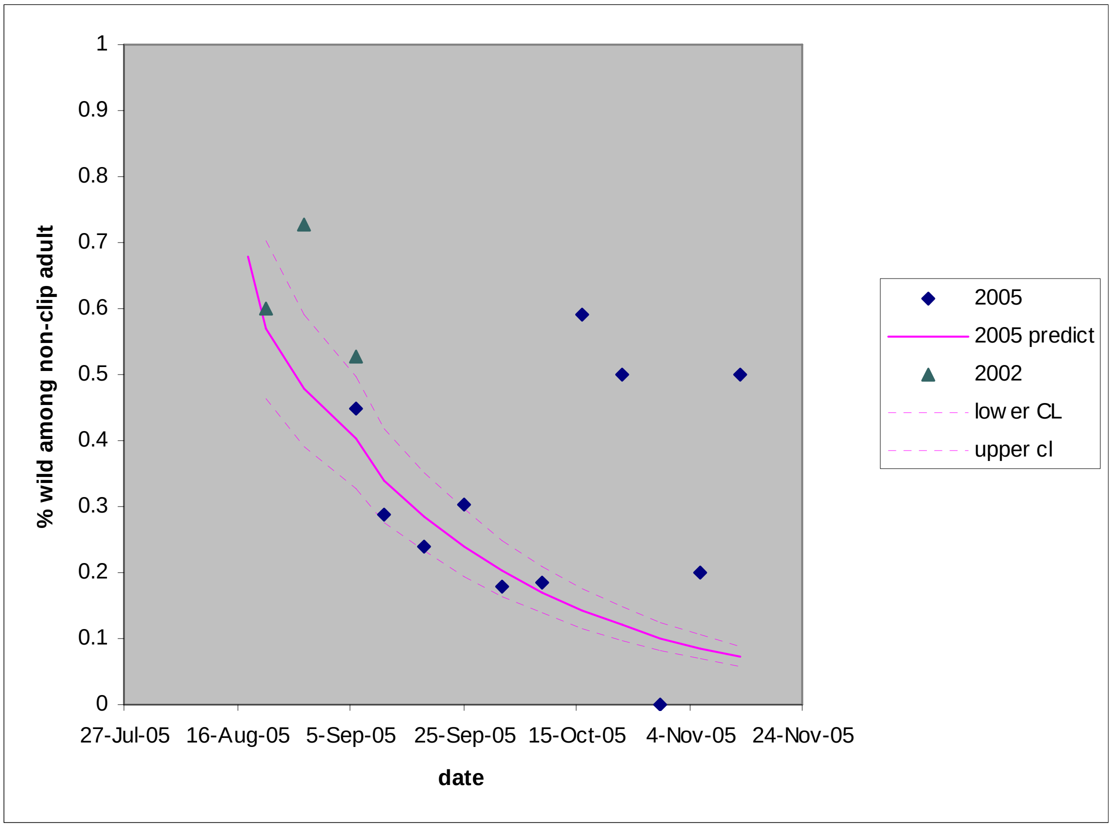
The group proportions were also applied to fish arriving before and after trapping with the exception of determining numbers of wild fish. Wild fish arrive at the dam earlier than hatchery fish in general (Figure 1). Expanding the wild fish proportions from the trapping period to pre- and post-trapping window counts would not accurately portray the run timing of wild fish.

The numbers of wild fish arriving during pre- and post-trapping was estimated by fitting an exponential regression to the % wild among unmarked fish during trapping and projecting the percentages to the weeks before and after trapping. The percent wild among unmarked fish was determined from scale samples taken from approximately 2 out of 3 trapped fish. Given these estimates of % wild adults and jacks before and after trapping by week, the numbers of wild adults and jacks were found by multiplying the expanded window counts of unmarked adults and jacks by the percentage.

After subtracting the numbers of wild adults and jacks pre- and post-trapping, the group proportions were applied to the remaining fish to complete the overall run reconstruction. The result was an estimate of numbers of fish returning to Lower Granite dam by group. In 2005, the total number of fall Chinook returning was estimated to be 13985 with a 90% confidence interval of (13434, 14523).

The literature review reported in the Phase 1 report described various approaches to run reconstruction, but no other run reconstruction process was found that parallels this one. The detailed window count, trap, and processing data sets available for this study were unique as far as we can determine. In 2006, Flynn, Punt, and Hilborn published a paper in the Canadian Journal of Fisheries and Aquatic Sciences describing a run reconstruction method for Bristol Bay sockeye. That paper described methods for estimating run timing more than run composition.

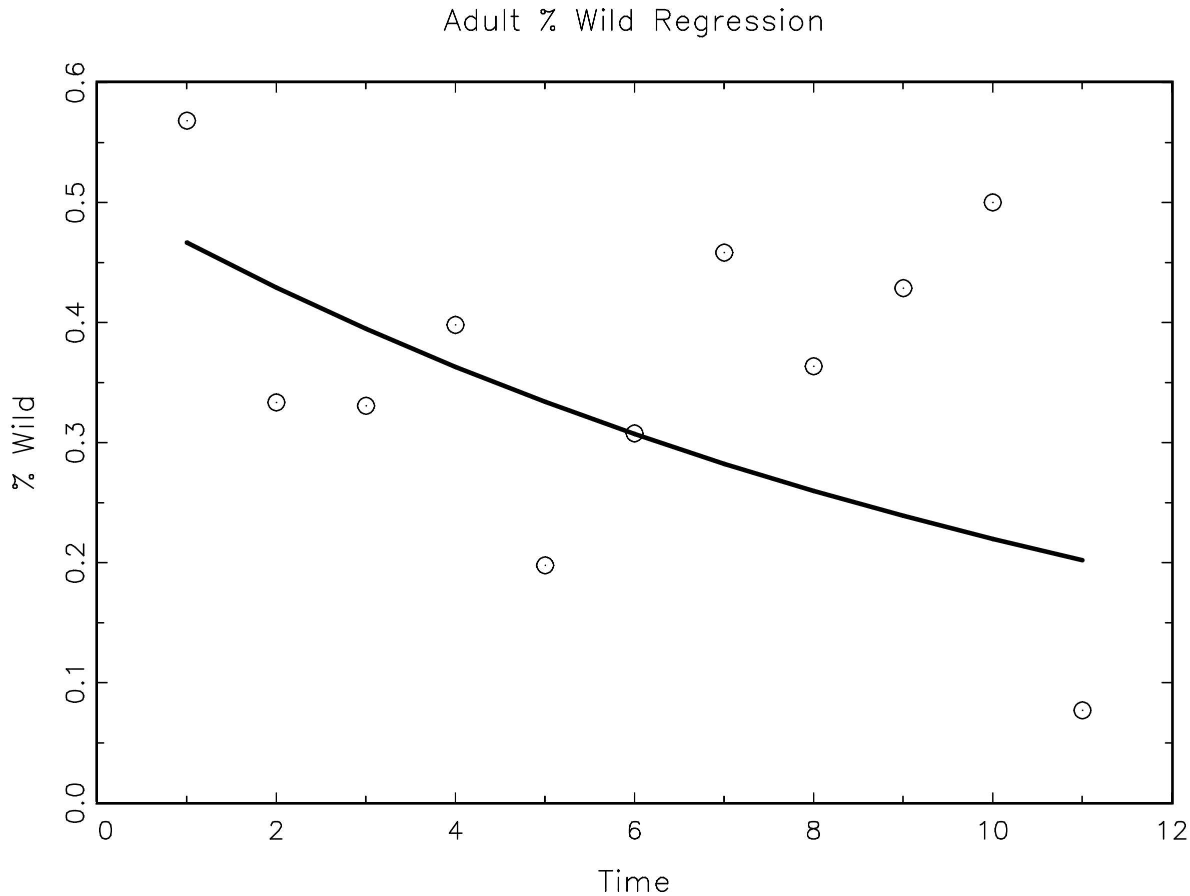
Figure 1. Regression of % wild versus week for adults for 2005 (from Henry Yuen, USFWS).



II. Validation of the bootstrap calculations

The run reconstruction process for the 2006 run was similar to the process used for the 2005 run with a few exceptions. The pre-trap period in 2006 ran from August 18th to August 31st. The trap was run from September 1st to November 21st. The post-trapping period was November 22nd to December 15th. For 2006 we partitioned adults into female and male so 3 sex categories (F,M,Jack) were used. Additionally, we provide confidence intervals for aggregates of groups as well as individual groups.

Figure 2. Exponential decay regression of proportion wild among unmarked adults, 2006.



The regressions for unmarked fish for 2006 confirm that the proportion of wild fish among unclipped fish declines as the season progresses (Figures 2 and 3). The details of the regression calculations for percent of wild fish are detailed on the Wild trap tab of LGRfallchinook.xls. The two points in the upper right corner of the adult regression plot were proportions based on only 4 and 7 fish and were omitted from the regression. The numbers of estimated wild fish before the trapping period was 218 adults and 84 jacks. The numbers of estimated wild fish after the trapping period was 2 adults and 0.3 jacks. These numbers were relatively minor when compared to the total number of wild fish in the run (3744).

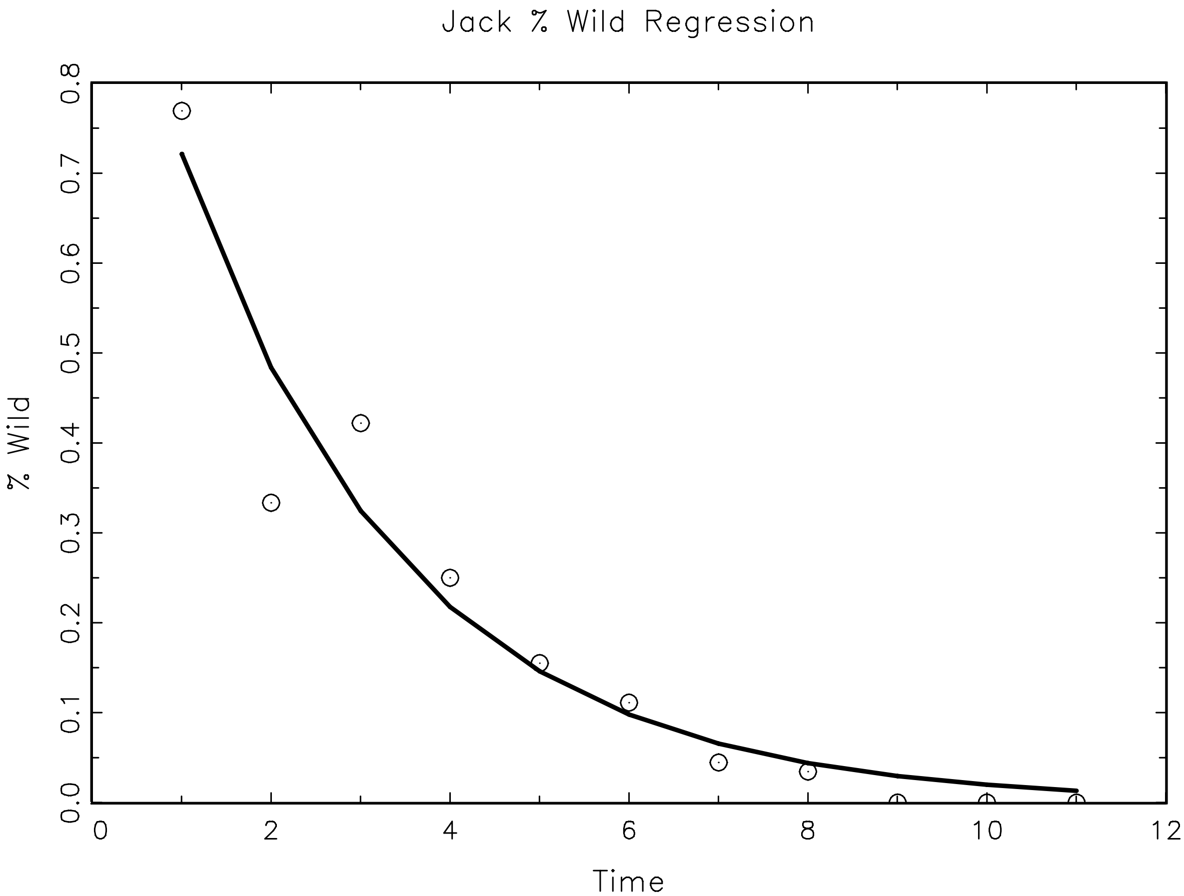
The inputs required to run the GAUSS program were:

Window counts—daily numbers of nonclipped and clipped adults and daily numbers of nonclipped and clipped jacks. Minijacks (fish <30 cm) were not counted at the window.

Trap data—date, week, sex, clip/noclip, coded wire tag presence or absence, PIT tag presence and number, and origin by scale sample. Note that the length of the fish was used to verify adult, jack, and minijack status, but was not read into the program. Minijacks were not included in the analysis.

Estimated Run Composition—numbers of females, males, and jacks for each group. Pre- and post-trapping composition combined with run estimate during trapping.

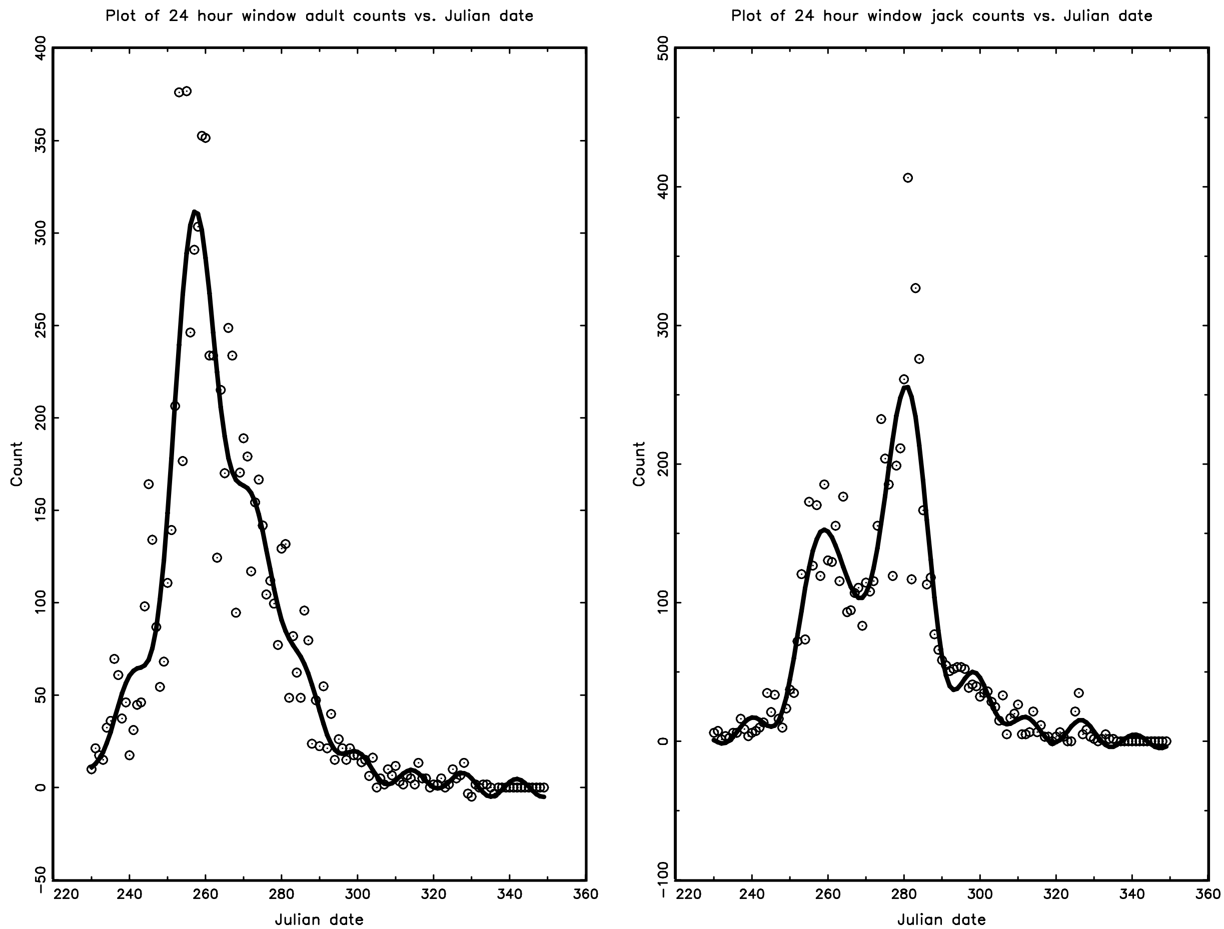
Figure 3. Exponential decay regression of proportion wild among unmarked jacks, 2006.



The bootstrap calculations as described in the Phase 1 report consisted of three parts. First a nonparametric bootstrap was applied to the window counts. This was done by fitting a model to the daily counts for adults and jacks, taking bootstrap samples of the residuals, and producing bootstrapped counts by adding the bootstrapped residuals to the daily model values. The model used is a 15 term fast Fourier transform (Figure 4). The bootstrapped daily counts were summed to obtain pre- and post-trapping estimates of numbers of adults and jacks arriving at the dam.

Second, a nonparametric bootstrap sample was taken from the trapping data base. In 2006, the number of fish trapped at the dam during the 13% trapping period was 1950. The estimate of numbers of fish arriving at the dam during the trapping period was 1950/0.13 = 15000. The number of bootstrap samples taken from the trapping database was obtained by generating a binomial random variable from a binomial with n=15000 and p = 0.13. The average number of bootstrap samples taken was 1950 with a standard deviation of 41.

Figure 4. Daily window counts and 15 term FFT model, 2006.



The bootstrap estimate of numbers of fish arriving at Lower Granite dam, N*, was the sum of the numbers of adults and jacks from the pre- and post-trapping bootstrap window counts and the number of bootstrap samples taken from the trapping database divided by 0.13.

Third, the composition of the bootstrapped run was found by taking a parametric multinomial sample for each of the N* fish. That is, we compute a multinomial trial N* times where the probabilities of a fish having come from any group was the proportion of that group in the 2006 run. Thus each fish was assigned a group at random in proportion to the percentages of each group in the run. In any bootstrap cycle, the composition will vary, but in general will follow the multinomial probability law governed by the group percentages calculated for this year's run.

1000 bootstrap samples were generated. Given the original data, we had 1001 values for 1) numbers of fish arriving pre-trap, 2) numbers of fish arriving during trapping, 3) numbers of fish arriving post-trap, and 4) the numbers of each group returning. By adding the pre-trap, trap, and post-trap numbers, we also had 1001 values for total fish returning to Lower Granite. By ordering each sequence of 1001 numbers, we computed the 90% lower and upper confidence interval for that quantity by locating the 5th and 95th quantile of the ordered list. For 2006, the values for numbers of fish returning appear in Table 1.

The confidence intervals for numbers returning overall and during trapping meet the goal of knowing the numbers of fish returning within 10% (the numbers in parentheses in Table 1 were 10% lower or higher than the estimates). The numbers of fish returning pre- and post- trapping were not known within 10%. The pre-trapping confidence interval target was 526 to 642. The calculated confidence interval of 475 to 695 was wider. The post-trapping confidence interval target was 73 to 89. The calculated confidence interval was 27 to 142. Because they only make up 4% of the total estimate, their imprecision does not greatly affect the precision of the overall estimator. This year's analysis shows that the bootstrap confidence intervals were valid and should be useful to the researcher and manager.

Table 1. Numbers of fall Chinook arriving at Lower Granite dam, 2006.

Time period	**Estimate**	Lower CI (10%)	Upper CI (10%)
Pre-trapping	**584**	475 (526)	695 (642)
Trap period	**15000**	**14400** (13500)	**15469** (16500)
Post-trapping	**81**	27 (73)	142 (89)
2006 Season	**15665**	**15079** (14098)	**16117** (17232)

Finding confidence intervals for individual groups was a more challenging undertaking. The 2006 90% confidence intervals for groups appear in Table 2. There were 109 groups. Although the bootstrap algorithm produces confidence intervals for each group, the results were not as precise as desired. A few of the confidence interval endpoints were within 10 percent of the estimate. This implies that we know the true number for those groups within 10% with 90% confidence. Estimates whose confidence intervals were within 10% of the estimate were marked with an *. Those estimates whose confidence intervals were within 20% of the estimate were marked with an **. Return groups with around 300 returns were generally known within 10%. Return groups with around 100 returns were often known within 20%. Smaller return groups were hard to estimate.

Table 2. 90% confidence intervals by group for numbers of fall Chinook to Lower Granite 2006.

Stock	L	F	U		L	M	U		L	J	U
LF04SPL1-IPCPA 073336	0	**0**	0		0	**0**	0		10	**17**	24
BONN04YUMA 092039	0	**0**	0		0	**0**	0		11	**18**	25
UMA00SUMA 093255	0	**0**	0		3	**8**	13		0	**0**	0
UMA02SUMA 093759	4	**8**	13		0	**0**	0		0	**0**	0
UMA03S 094028	0	**0**	0		10	**16**	24		0	**0**	0
UMA03S 094030	0	**0**	0		10	**16**	23		0	**0**	0
BONN03YUMA 094053	0	**0**	0		4	**9**	14		0	**0**	0
09BLANK yrl age 5 09BLANK	16	**25**	33		0	**0**	0		0	**0**	0
LF04SIPCHC 100471	0	**0**	0		0	**0**	0		5	**9**	14
LF04SIPCHC 106676	0	**0**	0		4	**9**	14		5	**10**	15
LF04SIPCHC 106776	0	**0**	0		0	**0**	0		19	**28**	37
LF03SIPCPA 106973	4	**8**	13		4	**9**	13		0	**0**	0
LF04SIPCHC 107176	0	**0**	0		4	**9**	13		12	**19**	26
LF03SIPCPA 107976	0	**0**	0		4	**8**	13		0	**0**	0
LF04SIPCHC 109370	0	**0**	0		4	**9**	14		0	**0**	0
LF01SCJA 610105	35	**45**	56		0	**0**	0		0	**0**	0
LF01SCJA 610106	15	**22**	30		0	**0**	0		0	**0**	0
NPTH02SO1 610107	4	**8**	14		0	**0**	0		0	**0**	0
NPTH04SA 610108	0	**0**	0		7	**12**	19		0	**0**	0
NPTH02SLVA 610109	5	**9**	14		0	**0**	0		0	**0**	0
NPTH02SO2 610110	37	**48**	60		5	**10**	15		0	**0**	0
LF01YCJA 610118	10	**16**	23		0	**0**	0		0	**0**	0
LF01YBCA 610119	10	**16**	23		0	**0**	0		0	**0**	0
LF02SBCA 610122	33	**43**	54		0	**0**	0		0	**0**	0
LF02SPLA 610123	10	**17**	24		0	**0**	0		0	**0**	0
LF04YBCA 610144	0	**0**	0		0	**0**	0		209	**236****	262
LF03YBCA 610145	4	**8**	13		62	**77****	92		24	**33**	42
LF03YPLA 610146	0	**0**	0		86	**102****	118		23	**32**	42
LF03YBCA 610147	0	**0**	0		73	**89****	104		17	**25**	34
LF04YBCA 610148	0	**0**	0		0	**0**	0		305	**338***	369
LF03YPLA 610149	0	**0**	0		126	**147****	167		23	**32**	41
LF04YPA 610150	0	**0**	0		0	**0**	0		189	**213****	238
LF04YCJA 610151	0	**0**	0		0	**0**	0		779	**835***	883
LF04YCJA 610152	0	**0**	0		0	**0**	0		917	**975***	1027
LF04YPA 610153	0	**0**	0		0	**0**	0		289	**320***	351
LF04SCJA 610154	0	**0**	0		15	**22**	31		86	**102****	119
LF04SCCD 610155	0	**0**	0		4	**9**	14		26	**36**	45
LF03SBCA 612500	66	**81****	96		143	**165****	186		0	**0**	0
LF01SPLA 612501	10	**16**	23		0	**0**	0		0	**0**	0
LF02YPLA 612502	37	**49**	60		0	**0**	0		0	**0**	0
LF02YCJA 612503	38	**48**	59		4	**8**	13		0	**0**	0
LF04SBCA 612504	0	**0**	0		34	**44**	55		231	**261****	290

* within 10% with 90% confidence ** within 20% with 90% confidence

Table 2 continued	L	**F**	U		L	**M**	U		L	**J**	U
LF03SCJA 612600	73	**88****	105		135	**157****	177		0	**0**	0
LF01SBCA 612639	31	**41**	52		13	**21**	28		0	**0**	0
LF02SCJA 612654	18	**26**	34		0	**0**	0		0	**0**	0
NPTH02SLVA 612657	4	**9**	14		0	**0**	0		0	**0**	0
LF02YBCA 612659	10	**16**	23		17	**24**	33		0	**0**	0
NPTH04SA 612669	0	**0**	0		0	**0**	0		22	**31**	40
NPTH04SA 612672	0	**0**	0		0	**0**	0		32	**42**	52
NPTH03SA 612675	25	**34**	44		0	**0**	0		0	**0**	0
LF00SBCA 630271	8	**14**	20		0	**0**	0		0	**0**	0
LF00SPLA 630272	6	**11**	17		0	**0**	0		0	**0**	0
LF01SO 630890	10	**16**	23		3	**8**	13		0	**0**	0
LF00YO 631273	23	**32**	41		0	**0**	0		0	**0**	0
PRIEST01SCOL 631382	33	**43**	54		0	**0**	0		0	**0**	0
LF02SCCD 631391	0	**0**	0		4	**8**	13		0	**0**	0
LF02SO 631545	23	**32**	42		4	**8**	13		0	**0**	0
LF01YO 631585	66	**81**	97		0	**0**	0		0	**0**	0
LF03YO 631769	23	**32**	42		351	**387***	423		96	**115****	133
LF03YO 631770	16	**24**	32		368	**401***	435		38	**50**	61
LF03SO 631786	10	**16**	23		31	**42**	52		0	**0**	0
KLICK02S 631796	4	**8**	13		0	**0**	0		0	**0**	0
LF02YO 632167	106	**126****	143		39	**51****	63		0	**0**	0
LF03YO 632368	0	**0**	0		10	**16**	23		0	**0**	0
LF04SGRRD 632782	0	**0**	0		11	**17**	24		43	**55**	67
LF04SO 632787	0	**0**	0		4	**9**	14		47	**60**	73
LF04YO 633283	0	**0**	0		0	**0**	0		592	**638***	681
LF04YO 633284	0	**0**	0		0	**0**	0		638	**690***	733
BLANK yrl age 2 BLANK	0	**0**	0		0	**0**	0		52	**65**	79
BLANK yrl age 4 BLANK	10	**16**	23		4	**8**	13		0	**0**	0
BLANK yrl age 5 BLANK	23	**33**	43		4	**8**	13		0	**0**	0
Inbasin unm/untag hatchery sub age 2 by scales- est unassociated Nez Perce Tribal Hatchery	0	**0**	0		0	**1**	2		1	**4**	8
Inbasin unm/untag hatchery sub age 2 by scales- est unassociated Couse Creek Direct	0	**0**	0		3	**7**	12		30	**40**	51
Inbasin unm/untag hatchery sub age 2 by scales- est unassociated Grande Ronde Direct	0	**0**	0		9	**15**	21		68	**84****	99
Inbasin unm/untag hatchery sub age 3 by scales- est unassociated Pittsburg Landing (non IPC)	0	**0**	0		24	**33**	42		0	**2**	4
Inbasin unm/untag hatchery sub age 5 by scales- est unassociated Big Canyon 2nd release	124	**145****	165		23	**32**	41		0	**0**	0
Inbasin AD only hatchery sub age 2 by scales- est unassociated Hells Canyon IPC	0	**0**	0		6	**12**	18		69	**85****	101
* within 10% with 90% confidence ** within 20% with 90% confidence											

Table 2 continued	L	**F**	U		L	**M**	U		L	**J**	U
Inbasin unm/untag hatchery sub age 2 by scales-unknown release site	0	**0**	0		30	**41**	51		209	**233***	261
Inbasin unm/untag hatchery sub age 3 by scales- unknown release site	0	**0**	0		443	**483***	520		15	**23**	31
Inbasin unm/untag hatchery sub age 4 by scales- unknown release site	231	**261****	288		322	**356***	386		0	**0**	0
Inbasin unm/untag hatchery sub age 5 by scales- unknown relelase site	201	**226****	249		38	**49**	62		0	**0**	0
Inbasin AD only hatchery sub age 3 by scales (not IPC)- unknown release site	0	**0**	0		3	**7**	11		0	**0**	0
Presumed IPC-Inbasin AD only hatchery sub age 4 by scales	31	**42**	53		9	**16**	23		0	**0**	0
Presumed IPC-Inbasin AD only hatchery sub age 5 by scales	11	**17**	24		0	**0**	0		0	**0**	0
Presumed IPC-Inbasin AD only hatchery sub age 6 by scales	4	**8**	13		0	**0**	0		0	**0**	0
Unknown hatchery yrl age 3 unm/untag	0	**0**	0		1	**4**	7		3	**7**	11
Unknown hatchery yrl age 4 unm/untag	11	**17**	24		4	**9**	14		0	**0**	0
Unknown hatchery yrl age 5 unm/untag	4	**9**	14		4	**8**	14		0	**0**	0
Unknown hatchery yrl age 4 AD only	13	**20**	27		2	**6**	10		0	**0**	0
Unknown hatchery yrl age 5 AD only	10	**16**	23		0	**0**	0		0	**0**	0
Possible HSTRAY unm/untag sub age 2 scales	0	**0**	0		18	**25**	34		111	**130****	149
Possible HSTRAY unm/untag sub age 3 scales	104	**123****	142		326	**358***	390		3	**8**	13
Possible HSTRAY unm/untag sub age 4 scales	299	**332****	363		165	**188****	210		0	**0**	0
Possible HSTRAY AD only sub age 2 scales	0	**0**	0		4	**8**	13		5	**9**	15
Possible HSTRAY AD only sub age 4 scales	46	**58****	70		4	**8**	13		0	**0**	0
Possible HSTRAY AD only sub age 5 scales	3	**8**	13		0	**0**	0		0	**0**	0
PIT tag unm/untag inbasin hatchery sub age 4-Research near Couse Creek	4	**9**	14		0	**0**	0		0	**0**	0
PIT tag unm/untag inbasin hatchery res rear age 5-Research near Couse Creek	4	**9**	13		0	**0**	0		0	**0**	0
WILD res rear age 2 scales	0	**0**	0		0	**0**	0		334	**369***	399
WILD res rear age 3 scales	0	**0**	0		252	**282***	308		33	**44**	57
WILD res rear age 4 scales	353	**387***	418		262	**291***	319		0	**0**	0
WILD res rear age 5 scales	295	**326***	356		41	**53**	66		0	**0**	0
WILD res rear age 6 scales	19	**28**	36		0	**0**	0		0	**0**	0
WILD sub age 2 scales	0	**0**	0		18	**26**	35		476	**519***	557
WILD sub age 3 scales	35	**46**	56		161	**186***	207		0	**0**	0
WILD sub age 4 scales	457	**498***	532		261	**291***	318		0	**0**	0
WILD sub age 5 scales	208	**234***	260		57	**71***	84		0	**0**	0
PIT tag unm/untag inbasin late migrant age 5-WILD res rear scales	4	**9**	15		0	**0**	0		0	**0**	0
PIT tag unm/untag presumed inbasin (H or W) late migrant age 4-WILD res rear scales	4	**9**	14		4	**9**	14		0	**0**	0

* within 10% with 90% confidence ** within 20% with 90% confidence

While individual researchers are very interested in the returns for a particular group, managers are more likely to be interested in the bigger picture. We calculated confidence intervals for strays (out-of-basin), Snake River hatchery fish, and wild fish by age (Table 3). Confidence intervals for small numbers of fish were still imprecise. For example, the confidence interval for male out of basin subyearling age 3 fish is 24 to 43. The desired limits would be 30 to 36. For larger numbers of fish, the goal of knowing the true number of fish returning within 10% with a confidence of 90% was met. For example, for female Snake River hatchery subyearlings of age 3, the confidence interval was 319 to 382. The desired limits were 316 to 386. We were 90% confident that the true number of age 3 female Snake River hatchery subyearlings was within 10% of the estimate of 351.

Table 3. 90% intervals for groups collapsed into stray, Snake River hatchery, and wild, 2006.

Stock/rearing type/total age	L	F	U		L	M	U		L	J	U
out-of-basin hatchery subyearling age 3	0	0	0		24	33	43		0	0	0
out-of-basin hatchery subyearling age 4	10	16	23		0	0	0		0	0	0
out-of-basin hatchery subyearling age 5	32	43	54		0	0	0		0	0	0
out-of-basin hatchery subyearling age 6	0	0	0		4	8	13		0	0	0
out-of-basin hatchery yearling age 2	0	0	0		0	0	0		69	83	97
out-of-basin hatchery yearling age 3	0	0	0		7	13	19		3	7	11
out-of-basin hatchery yearling age 4	41	54	66		15	23	31		0	0	0
out-of-basin hatchery yearling age 5	67	82	97		10	17	24		0	0	0
OUT OF BASIN	**172**	**196**	**221**		**77**	**93**	**108**		**74**	**89**	**106**
Snake R. hatchery subyearling age 2	0	0	0		222	248	274		1182	1254	1311
Snake R. hatchery subyearling age 3	319	351	382		1198	1261	1322		24	32	42
Snake R. hatchery subyearling age 4	839	896	952		549	593	637		0	0	0
Snake R. hatchery subyearling age 5	507	546	583		93	110	127		0	0	0
Snake R. hatchery subyearling age 6	24	33	44		0	0	0		0	0	0
Snake R. hatchery yearling age 2	0	0	0		0	0	0		4100	4245	4381
Snake R. hatchery yearling age 3	52	65	78		1156	1220	1282		258	288	315
Snake R. hatchery yearling age 4	212	239	264		68	83	99		0	0	0
Snake R. hatchery yearling age 5	96	114	132		0	0	0		0	0	0
Snake R. hatchery yearling age 6	23	32	41		0	0	0		0	0	0
SNAKE R. HATCHERY	**2175**	**2276**	**2363**		**3390**	**3516**	**3635**		**5636**	**5818**	**5994**
Snake R. Wild reservoir reared age 2	0	0	0		0	0	0		335	369	400
Snake R. Wild reservoir reared age 3	0	0	0		253	282	311		33	44	55
Snake R. Wild reservoir reared age 4	361	396	427		271	300	330		0	0	0
Snake R. Wild reservoir reared age 5	303	336	365		41	53	65		0	0	0
Snake R. Wild reservoir reared age 6	19	28	36		0	0	0		0	0	0
Snake R. Wild subyearling age 2	0	0	0		18	26	35		477	519	552
Snake R. Wild subyearling age 3	34	46	57		162	186	207		0	0	0
Snake R. Wild subyearling age 4	459	498	534		262	291	321		0	0	0
Snake R. Wild subyearling age 5	207	234	258		56	71	84		0	0	0
SNAKE R. WILD	**1456**	**1536**	**1610**		**1144**	**1209**	**1268**		**880**	**931**	**984**

III. “What if” analysis and the program

Having verified that the GAUSS program was stable and robust by applying it to a second year’s data, we next ask if it was flexible enough to ask interesting questions about choices that can be made about counting, trapping, and processing. Since one can build any 3 data sets for the program as described above (window counts, trap data, and composition), any return and sampling protocol can be simulated easily.

For example, suppose the run was twice the size it was in 2005 or 2006. By combining all of the window counts for 2005 and 2006 and combining the trapping databases, we simulated a run that was the size of both years combined. Assuming the female/male ratios were the same for both years and the combined composition resembles the 2006 composition, we adopted the composition from 2006 for the composition data set. All of the estimates were within 10% of the truth with 90% confidence except for out-of-basin males and jacks (Table 4). For out-of-basin males the target confidence interval was 154 to 188. The calculated confidence interval was 149 to 194, slightly larger. For jacks the target confidence interval was 149 to 183. The calculated confidence interval was 144 to 184.

Table 4. 90% confidence intervals on *combined* data sets from 2005 and 2006.

GROUP	lower	Female	Upper		lower	Male	upper		lower	Jack	upper
OUT OF BASIN	328	**362**	394		149	**171**	194		144	**166**	185
SNAKE R. HATCHERY	4061	**4212**	4346		6295	**6506**	6681		10439	**10766**	11037
WILD	2728	**2843**	2946		2138	**2320**	2372		1637	**1724**	1797
GRAND TOTAL (F,M,J)	28191	**28987**	29586								

The program also has a second “what if” option built into it. The program allows an arbitrary trapping rate to be set and confidence intervals can be found using that trapping rate with the current trapping data.

Table 5. 90% confidence intervals with the trapping rate at 13%, 2006.

Group/rearing type/total age	lower	Female	upper		lower	Male	upper		lower	Jacks	upper
OUT OF BASIN	171	**196**	219		76	**93**	108		74	**89**	105
SNAKE R. HATCHERY	2161	**2276**	2372		3349	**3516**	3651		5569	**5818**	6013
WILD	1448	**1536**	1613		1134	**1209**	1271		872	**931**	987

The 2005-2006 trapping rate was 13%. Does it make a difference if the trapping rate was higher or lower? We compared 13% to 6% and 20% trapping rates for groups collapsed to out-of-basin, Snake R. hatchery, and wild. If the trapping rate was set to 6%, the out of basin confidence intervals were comparable to the results found when the actual trapping rate of 13% was used. The other confidence intervals were generally wider. For example, the confidence interval for the grand total was 14696 to 16293. The results obtained with the 13% sampling rate were 15079 to 16117.

Table 6. 90% confidence intervals if trapping rate was 6%, 2006.

GROUP	lower	female	Upper	Lower	Male	upper	lower	jack	upper
OUT OF BASIN	169	**196**	220	75	**93**	109	73	**89**	105
SNAKE R.HATCHERY	2110	**2276**	2391	3276	**3516**	3685	5442	**5818**	6064
WILD	1422	**1536**	1621	1111	**1209**	1278	856	**931**	990
GRAND TOTAL (F,M,J)	14696	**15665**	16293						

If the trapping rate was set to 20%, the confidence intervals should be shorter. In some cases they were and in some cases they were not. Underlined values in Table 7 were tighter under 20% sampling when compared to the same confidence limits under 13%. The most notable improvement in confidence interval width was for the grand total.

Table 7. 90% confidence intervals if trapping rate was 20%, 2006.

GROUP	lower	female	upper	lower	male	upper	lower	jack	upper
OUT OF BASIN	172	**196**	218	76	**93**	108	74	**89**	104
SNAKE R. HATCHERY	<u>2173</u>	**2276**	2366	<u>3378</u>	**3516**	<u>3624</u>	<u>5620</u>	**5818**	<u>5981</u>
WILD	1455	**1536**	<u>1606</u>	<u>1145</u>	**1209**	1274	<u>876</u>	**931**	985
GRAND TOTAL (F,M,J)	<u>15212</u>	**15665**	<u>16034</u>						

IV. Comparison of 2005 and 2006 results

The individual groups were not comparable in 2005 and 2006, but we could compare the confidence intervals for out-of-basin, Snake River hatchery, and wild groups for the two years. During 2005 we only had adult and jack estimates so female and male data from 2006 were collapsed to adults for comparisons in Table 8. The run estimate was 12% higher in 2006 (13985 vs. 15665).

Table 8. Comparison of 2005 and 2006 90% confidence intervals.

Group	L	Adults	percent	U	L	Jacks	percent	U
2005								
OUT OF BASIN	1286	**1367.1**	**11.7%**	1446	126	**148.1**	**6.3%**	168
Snake R. Hatchery	6876	**7162.4**	**61.5%**	7446	1784	**1880.4**	**80.1%**	1986
Snake R. Wild	2986	**3108.6**	**26.7%**	3252	288	**318.7**	**13.6%**	350
GRAND TOTAL (adults+jacks)	13434	**13985**	14523					
2006								
OUT OF BASIN	258	**288**	**3.3%**	317	73	**89**	**1.3%**	105
Snake R. Hatchery	5593	**5792**	**65.6%**	5959	5624	**5818**	**85.1%**	5981
Snake R. Wild	2631	**2746**	**31.1%**	2847	873	**931**	**13.6%**	981
GRAND TOTAL (adults+jacks)	15079	**15665**	16117					

The proportions of Snake River hatchery fish were comparable for both years for both adults and jacks. The proportions of out-of-basin and wild fish were different for adults. There were relatively fewer out-of-basin adults in 2006 and more wild fish.

All but one of the confidence intervals in this table show that we know the true numbers of fish arriving at Lower Granite dam in these categories within 10%. For out-of-basin adults in 2006, the confidence interval was 258 to 317. The target confidence interval was 259 to 317. We were just outside the lower target. At the out-of-basin, hatchery, and wild level, sampling was adequate in both years.

In 2005 the coefficient of variation (CV) of the estimate was std error/estimate = 323.4/13985 = 2.3%. In 2006 the CV was 315.9/15665 = 2.0%. The precision for both years was comparable.

V. Discussion

The statistical intervals documented in this report were sensitive enough to detect differences in run sizes and gross composition in 2005 and 2006. The confidence intervals for total fish returning (13434 to 14523 and 15079 to 16117) do not overlap. The numbers of strays, Snake River hatchery, and wild fish can be seen to be different in the 2 years.

Interestingly, the proportion of jacks returning to Lower Granite dam was different in the two years. For 2005, the proportion was 2347/13985 = 17.5% and for 2006 the proportion was 6839/15665 = 43.6%. The confidence intervals for jacks do not overlap for the two years.

It was clear from the 2005 and 2006 analyses that we were not able to precisely estimate the numbers of fish returning by group in all cases. The confidence interval lower and upper limits for each group rarely were within 10% of the estimate. We do much better at the out-of-basin, hatchery, wild/age level (Table 3). We do even better at the out-of-basin, hatchery, wild level (Tables 5 and 8). We know the total numbers of fish returning within 10%.

In retrospect, constructing the run composition by female, male, and jack was a decided improvement in 2006. Since age at return and thus composition of females are different from males, the estimates were greatly improved by dividing adults in to females and males for the calculations. Managers need to know sex as well as age, The drawback to this separation of adults into female and male is that some of the individual numbers were smaller and, hence, our confidence intervals did not meet the 10% target. Where the numbers were adequate, however, we did a good job of estimating returns by sex and age.

References

Flynn, Lucy, Andre Punt, and Ray Hilborn (2006) A hierarchical model for salmon run reconstruction and application to the Bristol Bay sockeye salmon (*Oncorhynchus nerka*) fishery. Can. J. Fish. Aquat. Sci.63: 1564-1577.

Steinhorst, Kirk, Deborah Milks, and Bill Arnsberg (2006) Statistical Analysis of 2005 Lower Granite Dam fall Chinook Run Reconstruction. Report to the Pacific Salmon Commission Southern Boundary Restoration and Enhancement Fund, 15 pp.

APPENDIX A. Program listing

The GAUSS program consists of a main program (RR2006) and several subroutines. In addition the program uses several utilities for printing matrices and reading in input that were not listed here.

****

RR2006

****

```
/*  2007 version 2.1  8/8/07
    Bootstrap CI for total run to Lower Granite


    Developed for WDFW and NPT
    by Kirk Steinhorst at the University of Idaho
    Copyright 2007

    Declare global(s)

    cntr counts lines for printmat

*/
clear cntr,N1,N2,N3,k1,k2,k3,k4,f1,f2,f3,abeta,jbeta,total2,sss,flags;
library pgraph, cml;
graphset;
pqgwin many;
#INCLUDE cml.ext;
CMLset;
#include Getpara.g;
#include Getdat.g;
#include Getfnm.g;
#include Getalpha.g;
#include Getoptns.g;
#include Gettitle.g;
#include Printhd.g;
#include Printmat.g;
#include Printmix.g;
#include Printmsg.g;
#include Countmod.g;
```

```
#include TrapCalcs.g;
#include TrapWild.g;
#include Unmark.g;
#include Awild.g;
#include Jwild.g;
#include StarData.g;
#include StarComp.g;

/************************************************/
/*   MAIN --this is the main calling routine     */
/************************************************/
CLS;
CLOSEALL;

call printhd;
sss=0;
N1 = 14;  @ Number of days pre-trap @
N2 = 82;  @ Number of days in the trapping period @
N3 = 24;  @ Number of days post-trap @
k1 = 0.965; @ constant to convert from 16 to 24 hr window counts @
k2 = 5/6; @ Counts are 50 out of every 60 minutes @
k3 = 10/16; @ 10 hour video to 16 hour visual constant @
k4act = 0.13;  @ Actual trapping rate @
k4alt = 0.20;  @ "What if" value for trapping rate @
f1 = k1.*k2;
f3 = k3.*k1;
abeta = {1,1};
jbeta = {1,1};

@                     READ INPUT                          @

@ Get run details  @

infile = getfnm;
titl = Gettitle(infile);
outfile = infile $+ ".out";
OUTPUT file = ^outfile reset;
datestr(0); timestr(0);
call printmsg("RR2006 2.1 August 2007",1,0,0);
call printmsg(titl,1,0,0);
call printmsg( "Name of input file: " $+ infile,1,0,0);
```

```
call printmsg( "Name of output file: " $+ outfile,1,0,0);
alpha = getalpha(infile);
call printmat(alpha,"Alpha is");
flags = getoptns(infile);
temp = flags[1:4,1]';
call printmat(temp,"OPTIONS: model, printing, bootstrap");
IF flags[2,1] $=="detail"; flags[2,1] = "D"; ENDIF;
IF upper(strsect(flags[3,1],1,3)) $== "BOO"; flags[3,1] = "B";
ELSE; flags[3,1] = "N"; ENDIF;
B = flags[4,1];

@ Read in raw window counts from Steve    LGRWIN @

@ Spreadsheet columuns: date week adultnon adultclip jacknon jackclip @
LGRwinDat = SpreadsheetReadM("LGRfallchinook.xls","a8:f127",1);
WinDate = dttodtv(LGRwinDat[.,1]);
nwin = rows(LGRwinDat);
IF nwin /= (N1+N2+N3); call printmsg("nwin not equal to sum of N1, N2, N3: stopping",1,0,0); end; ENDIF;
call printmat(nwin,"Number of window count days");
ii = 1;
do while ii <= nwin;
  if ii == 1; WJulian = dayinyr(WinDate[ii,1:3]');
  else; WJulian = WJulian|dayinyr(WinDate[ii,1:3]');
  endif;
  ii = ii+1;
endo;


@ Final order of window data columns is week julian adultnon adultclip jacknon jackclip @
LGRwinDat = LGRwinDat[.,2]~WJulian~LGRwinDat[.,3:6];
IF flags[2,1] $== "D";
  call printmat(LGRwinDat,"Window count data: Week JulianDay Anon Acl Jnon Jcl"); ENDIF;

@ Read in LGR trap data @
@ The columns of interest are 1:date 2:week 7:sex 9:clip  10:wire 14:origin  @

TrapData = SpreadsheetReadSA("LGRTrap.xls","a2:n1960",1);
temp = rows(TrapData)~cols(TrapData);

@ Read in composition --uncomment the one you want @
```

```
@ CompNum = SpreadsheetReadM("Comp06.xls","b2:d28",1); @
@ CompNum = SpreadsheetReadM("Comp06.xls","b2:c4",2); @
CompNum = SpreadsheetReadM("Comp06.xls","b2:d4",3);
@ CompNum = SpreadsheetReadM("Comp06bygroup.xls","d2:f110",1); @

ng = rows(CompNum); nc = cols(CompNum);

@  Likewise uncomment the lines corresponding to the composition numbers spreadsheet above @

@ CompChar = SpreadsheetReadSA("Comp06.xls","a2:a28",1); @
@ CompChar = SpreadsheetReadSA("Comp06.xls","a2:a4",2); @
CompChar = SpreadsheetReadSA("Comp06.xls","a2:a4",3);
@ CompChar = SpreadsheetReadSA("Comp06bygroup.xls","a2:a110",1); @

@ Create a disk file for intermediate boostrap calculations of composition @

test = DeleteFile("AllComp.dat");
CREATE fa= AllComp with Col,nc.*ng,8;
test = close(fa);

call printmsg("END OF INPUT SECTION",1,0,0);

@                END OF INPUT                             @

@                ONE TIME CALCULATIONS                    @

@   First convert raw window counts to 24 hours     @
@   Video counts started on Nov 1 -- Julian Day 305 @

tem = ( (LGRwinDat[.,2] .< 305)./k1./k2 + (LGRwinDat[.,2].>= 305)./k1./k3 );
temp = LGRwinDat[.,3:6];
Win24 = tem.*temp; @ 24 hour counts for adultnon adultclip jacknon jackclip @

@ Find raw and adjusted weekly totals of unmarked adults and jacks from the window counts @
unmarked = Unmark(LGRwinDat[.,1]~Win24);

@ combine clipped and noclipped adults and jacks in window counts @
Wtotals = (Win24[.,1]+Win24[.,2])~(Win24[.,3]+Win24[.,4]);


/*  Remove summers found in trap from expanded window counts */
```

```
k4 = k4act;
Wtotals[15,1] = Wtotals[15,1] - 1/k4;
Wtotals[17,1] = Wtotals[17,1] - 1/k4;
Wtotals[18,1] = Wtotals[18,1] - 1/k4;
Wtotals[19,1] = Wtotals[19,1] - 1/k4;
Wtotals[20,1] = Wtotals[20,1] - 1/k4;
Wtotals[24,1] = Wtotals[24,1] - 3/k4;
Wtotals[30,1] = Wtotals[30,1] - 2/k4;
Wtotals[31,1] = Wtotals[31,1] - 2/k4;
Wtotals[36,1] = Wtotals[36,1] - 1/k4;
Wtotals[54,1] = Wtotals[54,1] - 1/k4;
@ Wtotals[30,2] = Wtotals[30,2] - 1/k4; coho @
@ Wtotals[34,2] = Wtotals[34,2] - 1/k4; coho @
@ Wtotals[36,2] = Wtotals[36,2] - 2/k4; coho @
@ Wtotals[37,2] = Wtotals[37,2] - 1/k4; coho @
@ Wtotals[39,2] = Wtotals[39,2] - 1/k4; steelhead @
@ Wtotals[44,2] = Wtotals[44,2] - 1/k4; coho @
@ Wtotals[47,2] = Wtotals[47,2] - 1/k4; coho @
Wtotals[55,2] = Wtotals[55,2] - 1/k4;
Wtotals[63,2] = Wtotals[63,2] - 1/k4;

@ Now set window data to week julian adult jack @

LGRwinDat = LGRwinDat[.,1:2]~Wtotals;

IF flags[2,1] $== "D";
  call printmat(LGRwinDat,"Expanded daily window counts--adults, jacks");

/* Window model for adults and jacks on a 24 hour basis */

  Wtotals = LGRwinDat[.,2 3]~zeros(nwin,1)~LGRwinDat[.,4]~zeros(nwin,1);

@ Estimate model for 24 hour window counts @
  Wtotals = Countmod(Wtotals,flags[1,1],"AJ");

@ Now do plot of window counts over time @
  begwind;
  window(1,2,0);
@  call printmat(Wtotals,"24 hour window counts after modelling--adults, jacks"); @
  setwind(1);
  X = Wtotals[.,1];
```

```
  Y = Wtotals[.,2:3];
  title("Plot of 24 hour window adult counts vs. Julian date");
  xlabel("Julian date");
  ylabel("Count");
  _plctrl = {-1,0};
  _pltype = {4,6};
  _pstype = {1,8};
  _pcolor = {0,1};
  _plwidth = 10;
  _pmcolor = {0,0,0,0,0,0,0,0,15};
  _ptek = "Window.tkf";
  xy(X,Y);
  nextwind;
  Y = Wtotals[.,4:5];
  title("Plot of 24 hour window jack counts vs. Julian date");
  xy(X,Y);
  endwind;

@ One time analyses of trapping data @
  {pJacks,Trapping,AgeOrign,UMbyWeek,unWno,HUWbyWeek} = TrapCalcs(TrapData);
  ntrap = rows(Trapping);
  TAdultsJacks = sumc(Trapping[.,2 3])';

@ Now do graph if requested @

/* Trapping model for adults and jacks on a 24 hour basis

First convert raw trap counts to 24 hours and add in zero columns as placeholders for modelled values */

  Trapping = Trapping[.,1]~Trapping[.,2]./k4~zeros(ntrap,1)~Trapping[.,3]./k4~zeros(ntrap,1);
  Trapping = Countmod(Trapping,flags[1,1],,"AJ");

  begwind;
  window(1,2,0);
  setwind(1);
  X = Trapping[.,1];
  Y = Trapping[.,2:3];
  title("Plot of Trap Adults vs. Julian date");
  xlabel("Julian date");
  ylabel("Count");
  _plctrl = {-1,0};
```

```
  _pltype = {4,6};
  _pstype = {1,8};
  _pcolor = {0,1};
  _plwidth = 10;
  _pmcolor = {0,0,0,0,0,0,0,0,15};
  _ptek = "Trapping.tkf";
  xy(X,Y);
  nextwind;
  Y = Trapping[.,4:5];
  title("Plot of Trap Jacks vs. Julian date");
  xy(X,Y);
  endwind;

ENDIF;

@  Start the bootstrap loop.  The 0-th iteration is for observed data. @

bb= 0;
cycle = 0;

DO WHILE bb <= B;

  cycle = cycle + 1;
  IF bb == 0; k4 = k4act;
  ELSE; k4 = k4alt; ENDIF;
  f2= k4;

@  Generate bootstrap samples for window counts and trap data @

  {LGRwin,Trapped} = StarData(LGRwinDat,TrapData,bb);

@ Do calculations on WINDOW counts for this cycle@

  Atau1hat = sumc(LGRwin[1:N1,3]);
  Jtau1hat = sumc(LGRwin[1:N1,4]);
  Ttau1hat = Atau1hat+Jtau1hat;
  Win1 = Atau1hat~Jtau1hat~Ttau1hat;
  Atau2win = sumc(LGRwin[(N1+1):(N1+N2),3]);
  Jtau2win = sumc(LGRwin[(N1+1):(N1+N2),4]);
  Ttau2win = Atau2win + Jtau2win;
  Win2 = Atau2win~Jtau2win~Ttau2win;
```

```
JpctWinTrap = Jtau2win/Ttau2win;
Atau3hat = sumc(LGRwin[(N1+N2+1):Nwin,3]);
Jtau3hat = sumc(LGRwin[(N1+N2+1):Nwin,4]);
Ttau3hat = Atau3hat+Jtau3hat;
Win3 = Atau3hat~Jtau3hat~Ttau3hat;
winEsts = Win1 + Win2 + Win3;

IF bb == 0;
  call printmsg("WINDOW ESTIMATES",1,0,0);
  call printmsg("**************************************************",1,0,0);
  call printmat(Win1,"Pretrap window estimates-Adult, jack, total");
  call printmat(Win2,"Window estimates during trapping period-Adult, Jack, and grand total");
  call printmat(Win3,"Post trap window estimates-Adult, jack, total");
  call printmat(winEsts,"Adult, jack and grand total estimates from WINDOW counts");
  call printmat(JpctWinTrap,"--->Percentage jacks in the window counts during trapping period");
  WinEst = winEsts;
  WinEst = WinEst~Win1;
  WinEst = WinEst~Win2;
  WinEst = WinEst~Win3;
  PtEst = (WinEst)';
  Win13 = Win1~Win3;
ELSE;
  WinEst = WinEst|(winEsts~Win1~Win2~Win3);
  Win13 = Win13|(Win1~Win3);
ENDIF;

@ Do calculations on TRAP counts for this cycle  @

  {pJacks,Trapping,AgeOrign,UMbyWeek,unWno,HUWbyWeek} = TrapCalcs(Trapped);

IF bb == 0;
  call printmsg("*********************************************",1,0,0);
  call printmsg("CALCULATIONS ON TRAPPING DATABASE",1,0,0);
  call printmsg("*********************************************",1,0,0);
  call printmat(pJacks,"Percent jacks in trap");
  call printmsg("Unmarked by week calculations",1,0,0);
  call printmsg("ACW indicates adult, clipped, wire ... ",1,0,0);
  call printmsg("JNNU means jack, nonclipped, nowire, unknown origin",1,0,0);
  call printmat(UMbyWeek,"Week ACW ACN ANW ANN ANNU JCW JCN JNW JNN JNNU by week_trap");
  call printmsg("Total unmarked adults and jacks--includes unmarked, no tag",1,0,0);
  call printmat(unWno,"Adult and jack total unmarked_trap");
```

```
    call printmsg("By week tally of hatchery, unknown, and wild origin",1,0,0);
    call printmat(HUWbyWeek,"Week AH AU AW JH JU JW by week_trap");


@  Now calculate numbers of wild fish @
    call printmsg("************************************************",1,0,0);
    call printmsg("CALCULATING NUMBERS OF WILD FISH PRE- AND  POST-",1,0,0);
    call printmsg("************************************************",1,0,0);

@ Find number wild during trapping using method 2 @
    AJtrpWld = TrapWild(AgeOrign[.,1],AgeOrign[.,2]);

@ Then estimate no. wild pre- and post-trapping @
    AWilds = AWild(HUWbyWeek,unWno,unmarked);
    JWilds = JWild(HUWbyWeek,unWno,unmarked);
    call printmsg("********************************************",1,0,0);
    call printmat(AWilds,"Adult wild pre- and post-trap");
    call printmat(JWilds,"Jack wild: pre- and post-trap");

  ENDIF;

@ Calculate fish to granite during period 2 using hybrid estimator @
  Trapcnt = sumc(Trapping[.,2 3])';
  Trapcnt = Trapcnt - (9*(1-pJacks)~9*pJacks); @ Adjust for extra fish and by-catch @
  tau2hat = trapcnt./k4;
  tau2hat = tau2hat~sumc(tau2hat');
  IF bb == 0;
    call printmsg("********************************************",1,0,0);
    call printmat(tau2hat,"Trap estimates for period 2-adult,jack,total");
    tauhat2 = tau2hat;
    total2 = tau2hat[1,3];
    PtEst = PtEst|tauhat2';
  ELSE;
    tauhat2 = tauhat2|tau2hat;
  ENDIF;

/* Now get hybrid estimate via window in periods 1 and 3 and trap during period 2
   and compute composition for this cycle                                        */

  tauhat = round(Win1[1,1:2]+tau2hat[1,1:2]+Win3[1,1:2]);
  tauhat = tauhat~sumc(tauhat');
```

```
@ Get bootstrap composition for this cycle @

  CompA = starComp(tauhat,CompNum,bb);

/* Multiply % composition by this cycle's estimate of number of fish returning to LGR
   ending up with numbers of fish for each stock--then store these numbers on disk  */
  Acompn = tauhat[1,3].*CompA';
  OPEN fa = Allcomp for append;
  test = writer(fa,Acompn);
  test = CLOSE(fa);

  IF bb == 0;
    PtEst = PtEst|tauhat';
    call printmsg("************************************************",1,0,0);
    call printmat(tauhat,"Hybrid estimates of season total to LGR-adult, jack, total");
    call printmsg("************************************************",1,0,0);
    PtEstComp = round(Acompn');
    hybrid = tauhat;
  ELSE;
    hybrid = hybrid|tauhat;
  ENDIF;

  bb = bb + 1;

ENDO; @ of bb @

@ If this is a bootstrap run, calculate confidence intervals @

IF B > 0;

@ First get confidence intervals for window and hybrid estimates of returning numbers @

  ThetaStar = WinEst~tauhat2~hybrid;
  B = rows(ThetaStar);
  call printmat(cycle~B,"Attempts and actual number of bootstrap samples");
  lndx = maxc(1|round((alpha/2) .* B));
  undx = round((1-alpha/2) .* B);
  se = stdc(ThetaStar);
  se = round(10.*se)./10;
  jj = 1;
```

```
  num_par = cols(ThetaStar);
  WinCI = zeros(num_par,2);
  DO WHILE jj <= num_par;
    ThetaStar[.,jj] = sortc(ThetaStar[.,jj],1);
    WinCI[jj,1] = ThetaStar[lndx,jj];
    WinCI[jj,2] = ThetaStar[undx,jj];
    jj = jj+1;
  ENDO;
  Ests = {"AWindow","JWindow","TotWin","AWin1","JWin1","TotWin1","AWin2","JWin2","TotWin2",
         "AWin3","JWin3","TotWin3", "ATrap","JTrap","TotTrap","AHybrid","JHybrid",
          "TotHyb"};

  PtEst = round(PtEst);
  WinCI = round(WinCI);
  call printmat(Ests~PtEst~WinCI~se,"Label     Estimate  Lower    Upper   Std Error");

  clear ThetaStar,LGRwin,Qry,Comb,Comp,AWildEst,JWildEst,hybrid,winest,tauhat2;

@ Now compute confidence intervals for composition @

  CLOSEALL;
  OPEN fa = Allcomp for read;
  rws = rowsf(fa);
  ntms = trunc(rws/100);
  jj = 1;
  DO WHILE jj <= nc.*ng;
    test = seekr(fa,1);
@ Get the jj column out of Allcomp external file @
    IF ntms == 0;
      results = readr(fa,rws);
      fish = results[.,jj];
    ELSE;
      kk = 1;
      DO WHILE kk <=ntms;
        test = seekr(fa,(kk-1).*100 + 1);
        results = readr(fa,100);
        IF kk == 1; fish = results[.,jj];
        ELSE; fish = fish|results[.,jj]; ENDIF;
        kk = kk+1;
      ENDO;
      IF ntms /= rws/100;
```

```
        test = seekr(fa,ntms.*100 + 1);
        rest = rws - test + 1;
        results = readr(fa,rest);
        fish = fish|results[.,jj];
      ENDIF;
    ENDIF;
    fish = sortc(fish,1);
    IF jj == 1;
      AgrpCI = fish[lndx,1]~fish[undx,1];
    ELSE;
      AgrpCI = AgrpCI|(fish[lndx,1]~fish[undx,1]);
    ENDIF;
    jj = jj+1;
  ENDO;
  CLOSEALL;
  AgrpCI = round(AgrpCI.* 10)./10;

  call printmsg("C O M P O S I T I O N",1,0,0);
  IF nc == 3;
    females = PtEstComp[1:ng]~AgrpCI[1:ng,.];
    males = PtEstComp[(ng+1):(2.*ng)]~AgrpCI[(ng+1):(2.*ng),.];
    jacks = PtEstComp[(2.*ng+1):(3.*ng)]~AgrpCI[(2.*ng+1):(3.*ng),.];
    temp =
females[.,2]~females[.,1]~females[.,3]~males[.,2]~males[.,1]~males[.,3]~jacks[.,2]~jacks[.,1]~jacks[.,3];
    call printmix(CompChar,temp,"GroupID  Female:  Male:  Jack: (LowerCI Estimate UpperCI)");
  ELSE;
    Adults = PtEstComp[1:ng]~AgrpCI[1:ng,.];
    Jacks = PtEstComp[(ng+1):(2.*ng)]~AgrpCI[(ng+1):(2.*ng),.];
    temp = Adults[.,2]~Adults[.,1]~Adults[.,3]~jacks[.,2]~jacks[.,1]~jacks[.,3];
    call printmix(CompChar,temp,"GroupID  Adult:  Jack: (LowerCI Estimate UpperCI)");
  ENDIF;

ENDIF; @ of IF B > 0 @

call printmsg("End of run",1,0,0);
datestr(0); timestr(0);
OUTPUT off;
end;
```

****
Awild.g
****

```
/* Estimation of pre- and post- wild for adults

   Called by main

   Calls ExpDecay

*/

PROC Awild(HUW,unmarkW,unmark);

  LOCAL W,temp,betahat,t,What,resid,SSresid,regrWild,APreWild,APostWild,wt,
        x,y;

@ Adults @
  wt = {1,1,1,1,1,1,1,1,0,0,1};
  temp = HUW[.,4]~unmarkW[.,1]~wt;
  W = (temp[.,1]./temp[.,2])~temp[.,3];
  W = missrv(W,0);
  betahat = ExpDecay(W);
  IF maxc(betahat) == 0; RETP(0~0); ENDIF;
  t = seqa(-1,1,rows(W)+6);
  What = betahat[1].*exp(-betahat[2].*t);
  What = recode(What,What .> 1,1);
  resid = ( W[.,1] - What[3:13]).*W[.,2];
  SSresid = resid'*resid;
  regrWild = unmark[.,1].*What;
  APreWild = sumc(regrWild[1:2]);
  APostWild = sumc(regrWild[14:17]);
  IF flags[2,1] $== "D";
    call printmat(HUW[.,4]~unmarkW[.,1]~wt,"Data for regression (adult)(AWild)");
    call printmat(W,"Percent wild and weights--adult(AWild)");
    call printmat(betahat,"betahats for adults(AWild)");
    call printmat(t~seqa(1,1,17)~What,"Predicted proportion wild adults(AWild)");
    call printmat(resid,"residuals(AWild)");
    call printmat(SSresid,"Sum of squared residuals(AWild)");
    call printmat(unmark[.,1]~What,"window unmarked and regression percents");
    call printmat(regrWild,"Predicted wild adults(AWIld)");
```

```
@ Do rough plot @
    ylabel("% Wild");        /* Y axis label */
    xlabel("Time");           /* X axis label */
    title("Adult % Wild Regression");  /* main title */
    x = t[3:13];
    y = W[.,1]~What[3:13];
    xy(x,y);                 /* call to main routine */
  ENDIF;

  RETP(APreWild~APostWild);

ENDP;
```

****
CountMod.g
****

```
/* This routine determines how the trap and LGR window counts are modelled

   Currently it models adults and jacks separately.

   June 15, 2006

   Called by MAIN

   Input: trap or window counts
          method = Ten (if window counts) or MA or FFT
   Output: count matrix with model columns changed (if method ^= Ten)

*/

PROC Countmod(data,method,series);

  LOCAL  FFTno,n,raw,temp1,temp2,M,x,y,yfft,r,Ser,sl,fuzz,flot;

  fuzz = 1.e-10;
  FFTno = 10; @ Default number of terms in FFT smoother @
  M = upper(strsect(method,1,1));
  Ser = upper(strsect(series,1,1));
  n = rows(data);

  IF M $=="M";  @ Calculate a 5 day moving average @
    IF Ser $== "T"; raw = data[.,2];
    ELSE; raw = data[.,2]~data[.,4];
    ENDIF;
    temp1 = movingave(raw,5);
    temp2 = (raw[1,.])|( (raw[1,.]+raw[2,.]+raw[3,.])/3 )|
             temp1[5:n,.]|( (raw[n-2,.]+raw[n-1,.]+raw[n,.])/3 )|(raw[n,.]);
    data[.,3] = temp2[.,1];
    IF Ser $== "A"; data[.,5] = temp2[.,2]; ENDIF;

  ELSEIF M $=="F"; @ Use FFT @
    temp1 = upper(method); sl = strlen(temp1);
    IF sl >= 5;
```

```
      IF strsect(temp1,1,3) $== "FFT"; @ Assume last two characters are terms in FFT @
@  Now see if the two characters are numeric and greater than zero  @
            flot = stof( strsect(temp1,sl-1,2) );
            IF flot < fuzz OR flot > n;
              CALL printmsg("Number of FFT terms is either character or outside of (0,n).",1,0,0);
              CALL printmsg("Setting number to 10 ",1,0,0);
            ELSE;
              FFTno = flot;
            ENDIF;
      ENDIF;
    ENDIF;

    x = seqa(1,1,n);
    y = data[.,2]; @ Adults or Total @
    yfft = rfft(y);
    r = rows(yfft);
    yfft[(FFTno+1):r,.] = zeros(r-FFTno,1);
    temp1 = rffti(yfft);
    data[.,3] = temp1[1:n];
    IF Ser $=="A" ;
      y = data[.,4]; @ Jacks @
      yfft = rfft(y);
      r = rows(yfft);
      yfft[(FFTno+1):r,.] = zeros(r-FFTno,1);
      temp1 = rffti(yfft);
      data[.,5] = temp1[1:n];
    ENDIF;
  ELSE; call printmsg("Model method misspecified...stopping",1,0,0); end;
  ENDIF;

  RETP(data);

ENDP;
```

****
ExpDecay.g
****

```
/* This routine computes exponential decay fit of % wild

   July 14, 2006

   Called by Awild and Jwild

   Input: Matrix with proportions in column 1 and weights in column 2

*/

proc lnlk(b,z);
    local dev,s2;
    dev = z[.,1] - b[1] * exp(-b[2]*z[.,2]);
    s2 = dev'dev/rows(dev);
    retp(lnpdfmvn(dev,s2));
endp;

proc grdlk(b,z);
    local d,s2,dev,r;
    d = exp(-b[2]*z[.,2]);
    dev = z[.,1] - b[1]*d;
    s2 = dev'dev/rows(dev);
    r = dev.*d/s2;
    retp(r~(-b[1]*z[.,2].*r));
endp;

proc hslk(b,z);
    local d,s2,dev,r, hss;
    d = exp(-b[2]*z[.,2]);
    dev = z[.,1] - b[1]*d;
    s2 = dev'dev/rows(dev);
    r = z[.,2].*d.*(b[1].*d - dev)/s2;
    hss = -d.*d/s2~r~-b[1].*z[.,2].*r;
    retp(xpnd(sumc(hss)));
endp;
```

```
PROC ExpDecay(wilddata);

  LOCAL  temp1,temp2,n,t,fuzz,theta0,yx,thetahat,f0,gee,h,retcode,
         startv,x,g,cov;

@  call printmat(wilddata,"wilddata in ExpDecay");wait;@
  n = rows(wilddata);
  t = seqa(1,1,n);
  fuzz = 1.e-10;
  yx = wilddata[.,1]~t;
@ call printmat(yx,"yx in ExpDecay");wait;@


  startv = { .50, 1 };

  CMLset;
  __title = "analytical Hessian and analytical gradient";
  _cml_MaxIters = 1000;
  _cml_HessProc = &hslk;
  _cml_GradProc = &grdlk;
  _cml_Bounds = { 0 10, 0 10 };   /* parameters must be positive */
  __weight = wilddata[.,2];

  output off;
@  { x,f0,g,cov,retcode } = CMLprt(CML(yx,0,&lnlk,startv));@
  { x,f0,g,cov,retcode } = CML(yx,0,&lnlk,startv);
  IF retcode > 1; x = {0,0}; ENDIF;
  output on;
  retp(x);
ENDP;
```

```
************************************************************************
```

Jwild.g

```
************************************************************************
/* Estimation of pre- and post- wild for jacks

   Called by main

   Calls ExpDecay

*/

PROC JWild(HUW,unmarkW,unmark);

  LOCAL W,temp,betahat,t,What,resid,SSresid,regrWild,JPreWild,JPostWild,wt,
        x,y;

@ Jacks @
  wt = {1,1,1,1,1,1,1,1,1,1,1};
  temp = HUW[.,7]~unmarkW[.,2]~wt;
  W = (temp[.,1]./temp[.,2])~temp[.,3];
  W = missrv(W,0);
  betahat = ExpDecay(W);
  IF maxc(betahat) == 0; RETP(0~0); ENDIF;
  t = seqa(-1,1,rows(W)+6);
  What = betahat[1].*exp(-betahat[2].*t);
  What = recode(What,What .> 1,1);
  resid = ( W[.,1] - What[3:13]).*W[.,2];
  SSresid = resid'*resid;
  regrWild = unmark[.,2].*What;
  JPreWild = sumc(regrWild[1:2]);
  JPostWild = sumc(regrWild[14:17]);

  IF flags[2,1] $== "D";
    call printmat(HUW[.,7]~unmarkW[.,2]~wt,"Data for regression (jack)(JWild)");
    call printmat(W,"Percent wild and weights--jack(JWild)");
    call printmat(betahat,"betahats for jacks(JWild)");
    call printmat(t~seqa(1,1,17)~What,"Predicted proportion wild jacks(JWild)");
    call printmat(resid,"residuals(JWild)");
    call printmat(SSresid,"Sum of squared residuals(JWild)");
    call printmat(unmark[.,1]~What,"window unmarked and regression percents");
```

```
    call printmat(regrWild,"Predicted wild jacks(JWild)");
@ Do rough plot @
    ylabel("% Wild");        /* Y axis label */
    xlabel("Time");           /* X axis label */
    title("Jack % Wild Regression");  /* main title */
    x = t[3:13];
    y = W[.,1]~What[3:13];
    xy(x,y);                 /* call to main routine */
  ENDIF;

  RETP(JPreWild~JPostWild);

ENDP;
```

****

StarComp.g

****

```
/*  This proc takes the current cycle's number of adults and jacks and uses
    a parametric (multinomial) bootstrap to produce this cycle's composition.

    Called by main

    The inputs are
       Nhat=tauhat  -- note. tauhat[1,1] = number adults; tauhat[1,2] = number jacks
       comp=CompNum -- the composition for females, males, and jacks from Debbie's
                       final run
       iter=bb -- the bootstrap iteration ( 0 is the original data )

    The outputs are
       A -- the percent composition for females and males
       J -- the percent composition for jacks

    August 3, 2997
                                                                             */
PROC StarComp(Nhat,comp,iter);

   LOCAL Adultcomp,AdultSum,Acomp,JackSum,Jcomp,nA,nJ,n,Atemp,Jtemp,ii,A,
         Aunif,temp,J,Junif,Atot,Jtot,Allcomp,AllSum,nAll,numcol;

   numcol = cols(comp);
   IF numcol == 3;
     Allcomp = comp[.,1]|comp[.,2]|comp[.,3];
   ELSE;
     Allcomp = comp[.,1]|comp[.,2];
   ENDIF;
   AllSum = sumc(Allcomp[.,1]);
   Acomp = Allcomp./AllSum;

   IF iter > 0;

     nAll = Nhat[1,3];
     n = rows(Allcomp);
     Atemp = cumsumc(0|Acomp[1:(n-1),1]);
```

```
        ii = 1;
        A = 0;
        Aunif = rndu(nAll,1);

        DO WHILE ii <= nAll;
          temp = sumc(Aunif[ii,1] .>= Atemp);
          A = A|temp;
          ii = ii +1;
        ENDO;
        A = trimr(A,1,0);

        ii = 1;
        Acomp = 0;
        DO WHILE ii <= n;
          Acomp = Acomp|sumc(A .== ii);
          ii = ii+1;
        ENDO;
        Acomp = trimr(Acomp,1,0);
        Atot = sumc(Acomp);
        Acomp = Acomp/Atot;

      ENDIF;

      RETP(Acomp);

   ENDP;
```

****

StarData.g

****

```
/*  This routine produces bootstrap data samples for
        window counts
        trapping data

    Called by main

    For window counts, you have to decide where to break the series into
    fairly homogenous segments

    July 29, 2007
*/

PROC (2)=StarData(L,Q,iter);

   LOCAL r,starL,Data,AJpredct,resid,get,rstar,temp,one,two,three,four,
         s,starQ,starC,weak,
         Atemp,Aunif,Jtemp,Junif,ii,A,J,
         Atot,Jtot,starCp,n,Aprob,Jprob,sx,sex,nA,nJ,u,nu,tem,
         Adultcomp,AdultSum,JackSum;


   IF iter == 0;

     RETP(L,Q);  @ In the first cycle, just return the original data @

   ELSE;


@ Select bootstrap samples for window counts and trapping @

@ Start with window counts @
     r = rows(L);
     starL = zeros(r,4);
     starL[.,1] = L[.,1]; starL[.,2] = L[.,2];

     Data = L[.,2:3]~zeros(r,1)~L[.,4]~zeros(r,1);
     AJpredct = Countmod(Data,"FFT15","AJ");
```

```
@ Period 1 -- start of run @
     one = AJpredct[1:N1,.];
@ Adults @
     resid = one[.,2] - one[.,3];
     get = trunc( rndu(N1,1).* N1 ) + 1;
     rstar = resid[get];
     one[.,2] = one[.,3] + rstar;
@ Jacks @
     resid = one[.,4] - one[.,5];
     get = trunc( rndu(N1,1).* N1 ) + 1;
     rstar = resid[get];
     one[.,4] = one[.,5] + rstar;

@ Period 2 -- major part of run with quite variable counts @
     two = AJpredct[(N1+1):(N1+47),.];
@ Adults @
     resid = two[.,2] - two[.,3];
     get = trunc( rndu(47,1).* 47 ) + 1;
     rstar = resid[get];
     two[.,2] = two[.,3] + rstar;
@ Jacks @
     resid = two[.,4] - two[.,5];
     get = trunc( rndu(47,1).* 47 ) + 1;
     rstar = resid[get];
     two[.,4] = two[.,5] + rstar;

@ Period 3 -- after the major part of run @
     three = AJpredct[(N1+47+1):(N1+N2),.];
@ Adults @
     resid = three[.,2] - three[.,3];
     get = trunc( rndu(35,1).* 35 ) + 1;
     rstar = resid[get];
     three[.,2] = three[.,3] + rstar;
@ Jacks @
     resid = three[.,4] - three[.,5];
     get = trunc( rndu(35,1).* 35) + 1;
     rstar = resid[get];
     three[.,4] = three[.,5] + rstar;

@ Period 4 -- end of run @
```

```
     four = AJpredct[(N1+N2+1):(N1+N2+N3),.];
@ Adults @
     resid = four[.,2] - four[.,3];
     get = trunc( rndu(24,1).*24 ) + 1;
     rstar = resid[get];
     four[.,2] = four[.,3] + rstar;
@ Jacks @
     resid = four[.,4] - four[.,5];
     get = trunc( rndu(24,1).*24 ) + 1;
     rstar = resid[get];
     four[.,4] = four[.,5] + rstar;

@ Put it together @
     temp = one|two|three|four;
     starL[.,3 4] = temp[.,2 4];

@ Select bootstrap sample of Qry: assume sample size fixed by Bernoulli sampling of Nhat, 0.13 @

     r = rows(Q);
     weak = {"3","4","5","6","7","8","9","10","11","12","13"};
@     s = r.*F2./K4; @

@ Set the number of samples in Q to binomial(total2,0.13)times by 0.13 @

     nu = total2; @ call printmat(nu,"Number of uniforms generated"); wait; @
     u = rndu(nu,1);
     s = sumc(u .<= k4);
     IF s < 11; s = 11; ENDIF;
     sss = sss|s;

     get = trunc( rndu(s,1).*r ) + 1;
     starQ = Q[get,.];
@ Set the first 11 randomly selected observations to different weeks, thus guaranteeing
  that each week has at least one observation @
     starQ[1,2] = cvtos(weak[1]);
     starQ[2,2] = cvtos(weak[2]);
     starQ[3,2] = cvtos(weak[3]);
     starQ[4,2] = cvtos(weak[4]);
     starQ[5,2] = cvtos(weak[5]);
     starQ[6,2] = cvtos(weak[6]);
     starQ[7,2] = cvtos(weak[7]);
```

```
      starQ[8,2] = cvtos(weak[8]);
      starQ[9,2] = cvtos(weak[9]);
      starQ[10,2] = cvtos(weak[10]);
      starQ[11,2] = cvtos(weak[11]);
      RETP(starL,starQ);

    ENDIF;

ENDP;
```

```
************************************************************************
```

TrapCalcs.g

```
************************************************************************
/*  This routine does all of the detailed calculations needed on the trap
    data base--especially for calculating the wild regressions

    Called by main

    The input is the trapping data base.
    The output consists of tables modelled after Henry Yuen's tabs in "HenryWild11".

    Note. Requires the Julian day for first and last trap day to be changed in the code.

    July 28, 2007
                                                                                 */
PROC (6) = TrapCalcs(Qry);

   LOCAL r,c,ii,age,cl,wi,we,orig,t,te,temp,tt,tem,sex,clip,wire,week,origin,
         weeks,rw,MUwk,HUWweek,Tadults,Tjacks,pctJacks,Atotun,Jtotun,jultemp,sl,jj,
         mon,da,yeer,place,pice,ymd,julian,days,rd,TAJ,TrapAJ,w,twelve,fmat;
   LOCAL AJtrpWld,sexx,dat,ag;

   r = rows(Qry);
   c = cols(Qry);
   week = strtrimr(strtriml(Qry[.,2]));
   sex = strtrimr(strtriml(Qry[.,7]));
   cl = strtrimr(strtriml(Qry[.,9]));
   wi = strtrimr(strtriml(Qry[.,10]));
   orig = strtrimr(strtriml(Qry[.,14]));

   ii = 1;
   DO WHILE ii <= r;

@ First find julian day @
     IF Qry[ii,1] $=="";
       jultemp = 0;
     ELSE;
       sl = strlen(Qry[ii,1]);
       jj = 1;
       mon = 0; da = 0; yeer = 0; place = 0;
```

```
        DO WHILE jj <= sl;
          pice = strsect(Qry[ii,1],jj,1);
          IF pice $== "/";
            IF place == 0; place = jj;
            ELSE; place = place|jj;
            ENDIF;
          ENDIF;
          jj = jj + 1;
        ENDO;
        mon = strtof(strsect(Qry[ii,1],1,place[1]-1));
        da = strtof(strsect(Qry[ii,1],place[1]+1,place[2]-1-place[1]));
        yeer = strtof(strsect(Qry[ii,1],place[2]+1, sl-place[2]));
        ymd = yeer|mon|da|0;
        jultemp = dayinyr(ymd);
      ENDIF;
@ Then convert the other columns from string to character value @
      w = stocv(week[ii]);
      t = stocv(sex[ii]);
      IF t $/= "J"; t = "A"; ENDIF;
      te = stocv(cl[ii]);
      temp = stocv(wi[ii]);
      tem = stocv(orig[ii]);
      IF tem $== "NOSCALES"; tem = ""; ENDIF;
      IF tem $== "";
        IF te $== "NO" and temp $== "N"; tem = "U";
        ELSE; tem = "H"; ENDIF;
      ENDIF;
      IF tem $== "HSTRAY"; tem = "S"; ENDIF;

      IF ii == 1;
        dat = jultemp~w~t~te~temp~tem;
      ELSE;
        dat = dat|(jultemp~w~t~te~temp~tem);
      ENDIF;
@  Note: order of dat is Julian week age clip wire origin @

      ii = ii +1;
    ENDO;

    julian = dat[.,1];
    week = dat[.,2];
```

```
   age = dat[.,3];
   clip = dat[.,4];
   wire = dat[.,5];
   origin = dat[.,6];

@  Calculate percent jacks during the trapping season @

   Tadults = sumc(age.$=="A");
   Tjacks = sumc(age.$=="J");
   pctJacks = Tjacks./(Tadults+Tjacks);

@ Now find trap totals by day @

   days = unique(julian,1);
   rd = rows(days);
   ii = 244; @ In 2006 the first trap day was September 1, JD = 244. @
   DO WHILE ii <= 325; @ The last trap day was November 21, JD = 325; @
     temp = delif(age,julian ./=ii);
     IF scalmiss(temp); TAJ = 0~0;
     ELSE; TAJ = sumc((temp .$== "A")~(temp .$== "J"))'; ENDIF;
     IF ii == 244; TrapAJ = ii~TAJ;
     ELSE; TrapAJ = TrapAJ | (ii~TAJ); ENDIF;
     ii = ii + 1;
   ENDO;

@ Do weekly tables @
   weeks = unique(week,1);
   rw = rows(weeks);
   ii =1;
   DO WHILE ii <= rw;
     ag = delif(age,week./=weeks[ii]);
     cl = delif(clip,week./=weeks[ii]);
     wi = delif(wire,week./=weeks[ii]);
     orig = delif(origin,week./=weeks[ii]);
     t =  MUweek(ag,cl,wi,orig);
     temp = tallyHUW(ag,orig);
     IF ii == 1;MUwk = weeks[ii]~t; HUWweek = weeks[ii]~temp;
     ELSE; MUwk = MUwk|(weeks[ii]~t); HUWweek = HUWweek|(weeks[ii]~temp); ENDIF;
     ii = ii+1;
   ENDO;
   Atotun = MUwk[.,4]+MUwk[.,5]-MUwk[.,6];
```

```
   Jtotun = MUwk[.,9]+MUwk[.,10]-MUwk[.,11];

   RETP(pctJacks,TrapAJ,age~origin,MUwk,Atotun~Jtotun,HUWweek);

 ENDP;


 /*  This proc does a cross-tab of age vs. origin

     August 2, 2006
                                                                    */
 PROC tallyHUW(S,O);

   LOCAL tem,temp,te,AH,AS,AU,AW,JH,JS,JU,JW,
         A,J,ApctWld,JpctWld,AUW,JUW,AUH,JUH;

@ Collapse origin = H and S @
  O = recode(O,O.$=="S","H");

@ Adults first @
   tem = S .$== "A";
   temp = O .$== "H";
   AH = sumc( tem .* temp );
   tem = S .$== "A";
   temp = O .$== "U";
   AU = sumc( tem .* temp );
   tem = S .$== "A";
   temp = O .$== "W";
   AW = sumc( tem .* temp );
@ Then juveniles @
   tem = S .$== "J";
   temp = O .$== "H";
   JH = sumc( tem .* temp );
   tem = S .$== "J";
   temp = O .$== "U";
   JU = sumc( tem .* temp );
   tem = S .$== "J";
   temp = O .$== "W";
   JW = sumc( tem .* temp );

   RETP(AH~AU~AW~JH~JU~JW);
```

```
 ENDP;

 /*  This proc finds marked and unmarked by week
     August 3, 2006                                */

PROC MUweek(S,C,W,O);

  LOCAL t,te,tem,temp,Adult,jack;

@ Adults first @
  t = S .$== "A";
  te = C .$== "AD";
  tem = W .$== "Y";
  Adult = sumc(t.*te.*tem);

  t = S .$== "A";
  te = C .$== "AD";
  tem = W .$== "N";
  Adult = Adult~sumc(t.*te.*tem);

  t = S .$== "A";
  te = C .$== "NO";
  tem = W .$== "Y";
  Adult = Adult~sumc(t.*te.*tem);

  t = S .$== "A";
  te = C .$== "NO";
  tem = W .$== "N";
  Adult = Adult~sumc(t.*te.*tem);
  temp = O .$== "U";
  Adult = Adult~sumc(t.*te.*tem.*temp);

@ Then jacks @
  t = S .$== "J";
  te = C .$== "AD";
  tem = W .$== "Y";
  Jack = sumc(t.*te.*tem);

  t = S .$== "J";
  te = C .$== "AD";
```

```
  tem = W .$== "N";
  Jack = Jack~sumc(t.*te.*tem);

  t = S .$== "J";
  te = C .$== "NO";
  tem = W .$== "Y";
  Jack = Jack~sumc(t.*te.*tem);

  t = S .$== "J";
  te = C .$== "NO";
  tem = W .$== "N";
  Jack = Jack~sumc(t.*te.*tem);
  temp = O .$== "U";
  Jack = Jack~sumc(t.*te.*tem.*temp);


  RETP( Adult~Jack );

ENDP;
```

```
***********************************************************************
```

TrapWild.g

```
***********************************************************************
/*  This proc computes wild during trapping using method 2 from HenryWild11
    on the calculate wild_LFH_NPTH tab

    Called by main during the 0 cycle--that is, for the real data cycle
*/

PROC TrapWild(S,O);
   LOCAL tem,temp,te,AH,AS,AU,AW,JH,JS,JU,JW,
         A,J,ApctWld,JpctWld,AUW,JUW,AUH,JUH, AtotWld,JtotWld,AtrapWld,JtrapWld;

@ call printmsg("Wild during trapping as per calculate wild LFH_NPTH tab",1,0,0); @

@ Calculate counts for Table 8 @
@ Adults first @
   tem = S .$== "A";
   temp = O .$== "H";
   AH = sumc( tem .* temp );
   tem = S .$== "A";
   temp = O .$== "S";
   AS = sumc( tem .* temp );
   tem = S .$== "A";
   temp = O .$== "U";
   AU = sumc( tem .* temp );
   tem = S .$== "A";
   temp = O .$== "W";
   AW = sumc( tem .* temp );
@ Then jacks @
   tem = S .$== "J";
   temp = O .$== "H";
   JH = sumc( tem .* temp );
   tem = S .$== "J";
   temp = O .$== "S";
   JS = sumc( tem .* temp );
   tem = S .$== "J";
   temp = O .$== "U";
   JU = sumc( tem .* temp );
   tem = S .$== "J";
```

```
  temp = O .$== "W";
  JW = sumc( tem .* temp );

  A = AH+AS+AU+AW;  call printmat(A,"Adults");
  ApctWld = AW/A;  call printmat(ApctWld,"Adult percent wild");
  AUW = AU*ApctWld;  call printmat(AUW,"Adult U that are wild");
  AUH = AU - AUW;  call printmat(AUH,"Adult U that are hatchery");
  AtotWld = AW + AUW;  call printmat(AtotWld,"Adults that are wild");
  Atrapwld = AtotWld/k4;  call printmat(AtrapWld,"Adult wild during trapping");
  J = JH+JS+JU+JW;  call printmat(J,"Jacks");
  Jpctwld = JW/J;  call printmat(JpctWld,"Jack percent wild");
  JUW = JU*JpctWld;  call printmat(JUW,"Jack U that are wild");
  JUH = JU - JUW;  call printmat(JUH,"Jack U that are hatchery");
  JtotWld = JW + JUW;  call printmat(JtotWld,"Jacks that are wild");
  Jtrapwld = JtotWld/k4;  call printmat(JtrapWld,"Jack wild during trapping");

  RETP(AtrapWld~JtrapWld);

ENDP;
```

****

Unmark.g

****

```
/*********************************************
  Calculates numbers of unmarked fish by week
  in the window

  Called by main
*********************************************/

proc Unmark(LGRw);

  LOCAL weeks,rw,ii,LGRwin_i,six8,temp,unmarkd;
  weeks = unique(LGRw[.,1],1);
  rw = rows(weeks);
  ii =1;
  DO WHILE ii <= rw;
    LGRwin_i = delif(LGRw,LGRw[.,1]./=weeks[ii]);
    temp = (sumc(LGRwin_i[.,2 4]))';
    IF ii == 1;unmarkd = temp;
    ELSE; unmarkd =unmarkd |temp; ENDIF;
    ii = ii+1;
  ENDO;
  RETP(unmarkd);
ENDP;
```

APPENDIX B. Example input data

****
Title and options—the file can have any name. The file name is read at run-time
****
title 2006 Run Reconstruction 3 stocks FMJ;
options output=ouch model=FFT bootstrap=1000;
alpha 0.1;

****
Window counts—note that these are RAW counts, not expanded
****

Date	week	Non-Clipped Adult	Clipped Adult	Non-Clipped Jack	Clipped Jack
18-Aug	1	6	2	4	1
19-Aug	1	12	5	3	3
20-Aug	1	10	4	1	0
21-Aug	1	13	-1	2	1
22-Aug	1	21	5	2	0
23-Aug	1	24	5	5	0
24-Aug	1	49	7	3	2
25-Aug	2	44	5	12	1
26-Aug	2	24	6	7	0
27-Aug	2	33	4	2	1
28-Aug	2	13	1	4	1
29-Aug	2	23	2	6	0
30-Aug	2	32	4	7	1
31-Aug	2	30	7	10	1
1-Sep	3	60	25	17	11
2-Sep	3	101	31	14	3
3-Sep	3	98	16	23	4

****

Trap data

****

TrapDate	Week	SpeciesAbbr	Run	SamplePct	REL/HAUL	Sexr	FL_cm	Clip	Wire	Recap?	PITTag	CWT	Origin_ScaleData
9/1/2006	3	CHIN	FALL	13	LFH	M	54	NO	N	FALSE			H
9/1/2006	3	CHIN	FALL	13	LFH	F	83	NO	N	FALSE			
9/1/2006	3	CHIN	FALL	13	LFH	F	84	NO	N	FALSE			W
9/1/2006	3	CHIN	FALL	13	REL	J	52	NO	N	FALSE			W
9/2/2006	3	CHIN	FALL	13	LFH	M	54	NO	N	FALSE			
9/2/2006	3	CHIN	FALL	13	LFH	M	66	NO	N	FALSE			W
9/2/2006	3	CHIN	FALL	13	LFH	M	69	NO	N	FALSE			
9/2/2006	3	CHIN	FALL	13	LFH	M	72	NO	N	FALSE			H
9/2/2006	3	CHIN	FALL	13	LFH	F	73	AD	N	FALSE			
9/2/2006	3	CHIN	FALL	13	LFH	F	77	NO	N	FALSE			H
9/2/2006	3	CHIN	FALL	13	LFH	F	80	AD	Y	FALSE			
9/2/2006	3	CHIN	FALL	13	LFH	F	86	NO	N	FALSE			W
9/2/2006	3	CHIN	FALL	13	LFH	F	94	NO	N	FALSE			W
9/2/2006	3	CHIN	FALL	13	LFH	F	96	NO	N	FALSE			W
9/2/2006	3	CHIN	FALL	13	REL	J	45	NO	N	FALSE			W
9/3/2006	3	CHIN	FALL	13	LFH	J	46	NO	Y	FALSE			
9/3/2006	3	CHIN	FALL	13	LFH	J	48	NO	Y	FALSE			
9/3/2006	3	CHIN	FALL	13	LFH	M	67	NO	N	FALSE			H
9/3/2006	3	CHIN	FALL	13	LFH	M	67	NO	N	FALSE			W
9/3/2006	3	CHIN	FALL	13	LFH	M	75	NO	N	FALSE			W
9/3/2006	3	CHIN	FALL	13	LFH	F	78	NO	N	FALSE			
9/3/2006	3	CHIN	FALL	13	LFH	M	78	NO	N	FALSE			
9/3/2006	3	CHIN	FALL	13	LFH	F	78	NO	N	FALSE			W
9/3/2006	3	CHIN	FALL	13	LFH	M	80	NO	N	FALSE			W
9/3/2006	3	CHIN	FALL	13	LFH	F	80	NO	N	FALSE			W
9/3/2006	3	CHIN	FALL	13	LFH	M	84	NO	N	FALSE			
9/3/2006	3	CHIN	FALL	13	LFH	M	85	NO	N	FALSE			
9/3/2006	3	CHIN	FALL	13	LFH	F	88	NO	N	FALSE			HSTRAY

****

Estimated run composition—this example was used to produce the CIs for Table 3

****

Origin_CWT	Females	Males	Jacks
out-of-basin hatchery subyearling age 3	0	33	0
out-of-basin hatchery subyearling age 4	16	0	0
out-of-basin hatchery subyearling age 5	43	0	0
out-of-basin hatchery subyearling age 6	0	8	0
out-of-basin hatchery yearling age 2	0	0	83
out-of-basin hatchery yearling age 3	0	13	7
out-of-basin hatchery yearling age 4	54	23	0
out-of-basin hatchery yearling age 5	82	17	0
Snake R. hatchery subyearling age 2	0	248	1254
Snake R. hatchery subyearling age 3	351	1261	32
Snake R. hatchery subyearling age 4	896	593	0
Snake R. hatchery subyearling age 5	546	110	0
Snake R. hatchery subyearling age 6	33	0	0
Snake R. hatchery yearling age 2	0	0	4245
Snake R. hatchery yearling age 3	65	1220	288
Snake R. hatchery yearling age 4	239	83	0
Snake R. hatchery yearling age 5	114	0	0
Snake R. hatchery yearling age 6	32	0	0
Snake R. Wild reservoir reared age 2	0	0	369
Snake R. Wild reservoir reared age 3	0	282	44
Snake R. Wild reservoir reared age 4	396	300	0
Snake R. Wild reservoir reared age 5	336	53	0
Snake R. Wild reservoir reared age 6	28	0	0
Snake R. Wild subyearling age 2	0	26	519
Snake R. Wild subyearling age 3	46	186	0
Snake R. Wild subyearling age 4	498	291	0
Snake R. Wild subyearling age 5	234	71	0

Addendum to:

Statistical Analysis of 2006 Lower Granite Dam Fall Chinook Run Reconstruction

Report for PSC Southern Boundary Restoration and Enhancement Fund Project:

Lower Granite Fall Chinook Run Reconstruction Assistance (Phase 2)

Kirk Steinhorst
Department of Statistics
University of Idaho
Moscow, ID 83844-1104
kirk@uidaho.edu

Deborah Milks
Washington Department of Fish and Wildlife
Snake River Lab
401 S. Cottonwood
Dayton, WA 99328
milksdjm@dfw.wa.gov

George Naughton
Department of Fish and Wildlife
University of Idaho
Moscow, ID 83844
naughton@uidaho.edu

and

Bill Arnsberg
Nez Perce Tribe
P.O. Box 365
Lapwai, ID 83540
billa@nezperce.org

August 20, 2007

Uncertainty related to origin of fish after CWT and unassociated assignments

Addendum 1 lists the final groupings of untagged hatchery fish that were not assigned to CWT or unassociated hatchery release groups. Each of the individual groups in Addendum 1 were included in Table 2, but the final assignments were not apparent from the name of the groups.

The “Inbasin unm/untag hatchery fish from unknown release sites” and “AD only hatchery fish from unknown release sites” (Addendum 1) were assigned to the Snake R. Hatchery group in Table 3. Since unassociated releases in the Snake R. did not have CWTs associated with them, SARs of fish from similar sites but from different return years were used to estimate their returns. It is possible that we underestimated the number of unassociated returns and the unm/untag fish in this group are actually from unassociated releases. In addition, because of uncertainty related to scale analysis, we suspect some proportion of these fish may be from out of basin, but we do not know to what extent.

The “Possible HSTRAY unm/untag sub” and “Possible HSTRAY AD only sub” groups were assigned to the Snake R. Hatchery group (Table 3) because CWT recoveries shed doubt on the magnitude of the estimated out of basin return using scale determinations. It is also possible that some of the Snake R. hatchery subyearlings reared at Umatilla Hatchery and Oxbow hatcheries (unassociated releases) were identified as HSTRAY by scale analysis. We do not have enough scale data from CWT fish from those hatcheries to determine if the scales have patterns similar to out of basin fish. Then again, it is possible that these fish were from out of basin unassociated releases, but we have no way of telling if that was the case.

The “Unknown hatchery yrl” were assigned to the out of basin group in Table 3. There is a lot of uncertainty regarding this group. Releases of yearling fall Chinook in the Snake R. basin have been essentially 100% marked or tagged over recent years. Yearling fall Chinook are also reared at Bonneville Hatchery but they are supposed to be 100% marked as well. Another option would be that the fish are untagged yearling releases of summer Chinook from the upper Columbia River. Scale analysis can only determine the age of the yearling hatchery fish, not origin so we rely solely on CWT data to determine the origin of yearlings.

There are many uncertainties related to scale analysis. Efforts are being made to continue to refine scale analysis for origin determinations. It is important that representative tag groups continue for each hatchery release to minimize uncertainty related to estimating the run composition at LGR.

Addendum 1. Presumed origins of untagged hatchery fish that could not be assigned to CWT or unassociated hatchery release groups.

Final groupings	Scale origins	Run Reconstruction Assignments	Individual groups (from Table 2)
Snake R. Hatchery	Inbasin hatchery	Presumed Inbasin (unknown hatchery)	Inbasin unm/untag hatchery sub age 2 by scales-unknown release site
			Inbasin unm/untag hatchery sub age 3 by scales-unknown release site
			Inbasin unm/untag hatchery sub age 4 by scales-unknown release site
			Inbasin unm/untag hatchery sub age 5 by scales-unknown relelase site
			Inbasin AD only hatchery sub age 3 by scales (not IPC)-unknown release site
		Presumed IPC	Presumed IPC-Inbasin AD only hatchery sub age 4 by scales
			Presumed IPC-Inbasin AD only hatchery sub age 5 by scales
			Presumed IPC-Inbasin AD only hatchery sub age 6 by scales
	Possible HSTRAY (hatchery)	Presumed Inbasin (unknown hatchery)	Possible HSTRAY unm/untag sub age 2 scales
			Possible HSTRAY unm/untag sub age 3 scales
			Possible HSTRAY unm/untag sub age 4 scales
		Presumed IPC	Possible HSTRAY AD only sub age 2 scales
			Possible HSTRAY AD only sub age 4 scales
			Possible HSTRAY AD only sub age 5 scales
Out of basin	Unknown hatchery	Presumed stray (unknown hatchery)	Unknown hatchery yrl age 3 unm/untag
			Unknown hatchery yrl age 4 unm/untag
			Unknown hatchery yrl age 5 unm/untag
			Unknown hatchery yrl age 4 AD only
			Unknown hatchery yrl age 5 AD only

Differences between determination of origins in 2005 and 2006 datasets

In 2005, HSTRAY fish were assigned out of basin, which is different from what was done for 2006 after discrepancies were noted in blind tests. Origins for other fish were assigned in the same manner each year. This report compares the statistical confidence intervals for out of basin, Snake River hatchery, and Snake River wild adults and jacks. Although the differences in estimates are partly due to differences in HSTRAY assignment, the assessment of precision as measured by the confidence interval widths is still relevant.