STUDY PERFORMANCE REPORT

State: Michigan

Project No.: F-81-R-7

Study No.: 230721

Title: Design, analysis, and implementation of aquatic resource inventory in Michigan.

Period Covered: October 1, 2005 to September 30, 2006

Study Objectives: (1) Assist in the continued design, analysis and reporting of a sampling plan for a statewide inventory of aquatic resources that Fisheries Division is responsible for.

Summary: My primary efforts this year have been directed toward developing data summarization tools for the resource inventory program. These programs were developed with in collaboration with Todd Wills and Kevin Wehrly to insure that the data structure and sampling design was appropriately accounted for in the analysis.

Findings: Jobs 2 through 6 were scheduled for 2005-06, and progress is reported below.

Job 2. Title: Develop and present training modules.–I presented the two-day continuing education course on the theoretical basis and application of statistical sampling I had developed in 2003 (and had presented in 2004) in February of 2005 and September of 2006 to a total of 42 participants.

Job 3. Title: Analyze statistical properties of proposed modifications–The paper submitted last year to Fisheries on this topic was revised, and subsequently published (citation provided under findings for Job 6). The paper in preparation last year was completed and submitted to Canadian Journal of Fisheries and Aquatic Sciences; the reprint will be provided with the annual performance report for 2006-07.

Job 4. Title: Assist in the development of analysis and reporting tools.–The SAS-based analysis programs for the appropriate analysis of resource inventory data have been extensively revised and tested. Copies of a representative sample of these programs are contained in Attachment A. Draft tables summarizing much of the lake sampling data have been prepared, and will be reported upon by Kevin Wehrly.

Job 5. Title: Assist in the production of a publication for the general public reporting the results of state-wide sampling.–Results of the analyses conducted under Job 4 have been incorporated into a preliminary outline of a report intended for the general public.

Job 6. Title: Prepare annual report and communicate program results.–This annual report was prepared as scheduled. Program results from the resource inventory were reported to MDNR biologists and technicians at their annual Trout Committee meeting, at several meetings attended by all fisheries division biologists, and at basin team meetings by MDNR fisheries research biologists A. Nuhfer, K. Wehrly, and T. Wills. I presented additional program information at Fisheries Division’s annual research section meeting in 2006.

The following Fisheries article contains findings for this study and is provided with the final report for F-80-R-7, Study 230714.

Wagner, T., Hayes, D. B. and Bremigan, M. T. 2006. Accounting for multilevel data structures in fisheries data using mixed models. Fisheries 31:180-185.

Prepared by: Daniel Hayes
Dated: September 30, 2006

Attachment A. Draft SAS programs for the analysis of resource inventory data.

```
OPTIONS PS=250 ls=200 nocenter nodate;

libname h_drive "H:\";

%let species="BLG";



* Read in Effort data.  This could be arranged differently, but this seems
to work ok.;
DATA EFFORT;
input survey year angling FYKENET   GLGNET      IGNET MINIFYKE    SEINE
      SHOCK TRAPNET;
cards;
4030  2003  -999  2     -999  1     2     2     -999  2

datalines deleted…
;
run;

data catches;
input survey gear $ spe_name $ catch;
cards;
3493  FYKENET     BLG   64

datalines deleted…
;
run;



*Select out target species;

data &species;
set catches;
if spe_name=&species;
run;

* Do sorts in preparation for merging catches and effort data;
proc sort data=effort;
by survey;
run;

proc sort data=&species;
by survey;
run;


*Merge effort and catch data;
data m;
merge effort &species;
by survey;
run;



* Take merged data, and set all missing catches to zero.  Also, reconfig
data so that catch is denoted separately for
each gear type;
```

```
data all_gear;
set m;

SHOCK=SHOCK/3600;

if gear="ANGLING" AND ANGLING>0 THEN ANGLING_cpue=CATCH/ANGLING;
if ANGLING>0 AND CATCH=. THEN ANGLING_cpue=0;

if gear="SHOCK" AND SHOCK>0 THEN SHOCK_cpue=CATCH/SHOCK;
if SHOCK>0 AND CATCH=. THEN SHOCK_cpue=0;

if gear="IGNET" AND IGNET>0 THEN IGNET_cpue=CATCH/IGNET;
if IGNET>0 AND CATCH=. THEN IGNET_cpue=0;

if gear="GLGNET" AND GLGNET>0 THEN GLGNET_cpue=CATCH/GLGNET;
if GLGNET>0 AND CATCH=. THEN GLGNET_cpue=0;

if gear="FYKENET" AND FYKENET>0 THEN FYKENET_cpue=CATCH/FYKENET;
if FYKENET>0 AND CATCH=. THEN FYKENET_cpue=0;

if gear="TRAPNET" AND TRAPNET>0 THEN TRAPNET_cpue=CATCH/TRAPNET;
if TRAPNET>0 AND CATCH=. THEN TRAPNET_cpue=0;

if gear="SEINE" AND SEINE>0 THEN SEINE_cpue=CATCH/SEINE;
if SEINE>0 AND CATCH=. THEN SEINE_cpue=0;

if gear="MINIFYKE" AND MINIFYKE>0 THEN MINIFYKE_cpue=CATCH/MINIFYKE;
if MINIFYKE>0 AND CATCH=. THEN MINIFYKE_cpue=0;

RUN;


* This summary is used to "flatten" the data set - meaning aggragating the
multiple records contained for each survey;
* into a single record;

PROC SUMMARY;
BY SURVEY;
ID YEAR;
VAR ANGLING_CPUE SHOCK_CPUE IGNET_CPUE GLGNET_CPUE FYKENET_CPUE
TRAPNET_CPUE SEINE_CPUE MINIFYKE_CPUE;
OUTPUT OUT=SUM1 MAX=;
RUN;

data strat;
set h_drive.stratum;
run;

proc sort;
by survey;
run;

data all;
merge sum1 strat;
by survey;
*format stratum $20;
if size_stratum="T" then stratum="Tiny "||depth_stratum;
if size_stratum="S" then stratum="Small "||depth_stratum;
if size_stratum="M" then stratum="Medium "||depth_stratum;
if size_stratum="L" then stratum="Large "||depth_stratum;
dummy="All";
```

```
run;

proc sort;
by stratum;
run;

proc print;
run;

*===========================================================================
====================================;
* Begin data summary and analysis;
*===========================================================================
====================================;

* This set of tables is for most disaggregated level;
PROC TABULATE format=7.2;
Class YEAR basin depth_stratum stratum;
VAR SHOCK_CPUE IGNET_CPUE FYKENET_CPUE TRAPNET_CPUE SEINE_CPUE
MINIFYKE_CPUE glgnet_cpue;
TABLE stratum all, basin* (SHOCK_CPUE ) * (MEAN n stderr)
all*(shock_cpue*(mean n stderr))/rts=20;
TABLE stratum all, basin* (IGNET_CPUE) * (MEAN n stderr)
all*(ignet_cpue*(mean n stderr))/rts=20;
TABLE stratum all, basin* (FYKENET_CPUE ) * (MEAN n
stderr)all*(fykenet_cpue*(mean n stderr))/rts=20;
TABLE stratum all, basin* (TRAPNET_CPUE ) * (MEAN n
stderr)all*(trapnet_cpue*(mean n stderr))/rts=20;
TABLE stratum all, basin* (SEINE_CPUE ) * (MEAN n
stderr)all*(seine_cpue*(mean n stderr))/rts=20;
TABLE stratum all, basin* (MINIFYKE_CPUE) * (MEAN n stderr )
all*(minifyke_cpue*(mean n stderr))/rts=20;
TABLE stratum all, basin* (GLGNET_CPUE) * (MEAN n stderr )
all*(glgnet_cpue*(mean n stderr))/rts=20;
RUN;

* This set of tables is focused on lake size differences;
PROC TABULATE format=7.2;
Class YEAR basin depth_stratum stratum size_stratum;
VAR SHOCK_CPUE IGNET_CPUE FYKENET_CPUE TRAPNET_CPUE SEINE_CPUE
MINIFYKE_CPUE glgnet_cpue;
TABLE size_stratum all, basin* (SHOCK_CPUE ) * (MEAN n stderr)
all*(shock_cpue*(mean n stderr))/rts=20;
TABLE size_stratum all, basin* (IGNET_CPUE) * (MEAN n stderr)
all*(ignet_cpue*(mean n stderr))/rts=20;
TABLE size_stratum all, basin* (FYKENET_CPUE ) * (MEAN n
stderr)all*(fykenet_cpue*(mean n stderr))/rts=20;
TABLE size_stratum all, basin* (TRAPNET_CPUE ) * (MEAN n
stderr)all*(trapnet_cpue*(mean n stderr))/rts=20;
TABLE size_stratum all, basin* (SEINE_CPUE ) * (MEAN n
stderr)all*(seine_cpue*(mean n stderr))/rts=20;
TABLE size_stratum all, basin* (MINIFYKE_CPUE) * (MEAN n stderr )
all*(minifyke_cpue*(mean n stderr))/rts=20;
TABLE size_stratum all, basin* (GLGNET_CPUE) * (MEAN n stderr )
all*(glgnet_cpue*(mean n stderr))/rts=20;
RUN;

* Begin calculating frequency statistics;
proc summary data=all;
var SHOCK_CPUE IGNET_CPUE FYKENET_CPUE TRAPNET_CPUE SEINE_CPUE
MINIFYKE_CPUE glgnet_cpue;
```

```
output out=freq1 p90=shock90 ignet90 fykenet90 trapnet90 seine90 minifyke90
glgnet90;
run;

data freq2;
set freq1;
shock_bin=shock90/7;
*Use this dummy variable to help merge these results back to all data set;
dummy="All";

run;


data a1;
merge all freq2;
by dummy;
run;

data a2;
set a1;
shock=int(shock_cpue/shock_bin)+1;
run;

proc sort;
by shock;
run;


proc summary;
by shock;
id shock_bin;
var shock_cpue;
output out=freq3 n=count;
run;


* Note - catches of zero are put in smallest bin - do we want to put them
in separate category?;
data freq4 (drop=shock_bin _type_ _freq_);
set freq3;
if shock=. then delete;
shock_min=shock_bin*(shock-1);
shock_max=shock_bin*shock;
run;

proc print;
run;
```

```
OPTIONS PS=250 ls=200 nocenter nodate;

libname h_drive "D:\S&T summaries\2006 summary work";

%let sp="YEP";      *Select 3-letter species code here;
%let max_len=18;   *Select maximum mean_length for frequency distributions
of mean lenghts;
%let len_step=0.5; *Select step size for bins for mean length frequency
distribtution;

/* Note: Default bin size for mean length frequency distributions is 0.5
inches.  This is hard-wired in below  */


data l;
set h_drive.lengths;

** Note: some observations have a count of zero for an inch group.  This
can lead to errors later on, but this needs to be investigated further
in the raw database Hayes and Wehrly June 2006;

if number=0 then delete;

run;

*Select out target species;

data &sp;
set l;
if species=&sp;
run;

proc sort data=&sp;
by survey;
run;


data strat;
set h_drive.stratum;
run;

proc sort;
by survey;
run;

data all;
merge &sp strat;
by survey;
if size_stratum="T" then stratum="Tiny "||depth_stratum;
if size_stratum="S" then stratum="Small "||depth_stratum;
if size_stratum="M" then stratum="Medium "||depth_stratum;
if size_stratum="L" then stratum="Large "||depth_stratum;
dummy="All";
run;


*===========================================================================
====================================;
* Begin data summary and analysis;
*===========================================================================
====================================;
```

```
* Compute product of number of fish in length group times it's midpoint to
calculate mean length;
data a1;
set all;
product=number * (inch_group+0.5);
run;

proc sort;
by gear survey;
run;

* Add up all these products and count total number of fish;
proc summary;
by gear survey;
id dummy;
var number product;
output out=sum1 sum=sum_number sum_product;
run;


data means;
set sum1;
mean_length=sum_product/sum_number;
run;



proc univariate normal plot ;
by gear;
var mean_length;
run;


data mean_l_bin;
set means;
mean_l_bin=int(mean_length*2)/2 +0.25;
run;

title "Mean length by survey by gear";
proc print;
run;


* Create data set, using trapnet as base to force zeros in frequency count
of mean lengths. This helps insure that all bin intervals occur in
following
proc tabulate;
data add_zeros;
do mean_l_bin=0.25 to &max_len by &len_step;
mean_length=.;
gear="TRAPNET";
output;
end;
run;

data mean_l;
set mean_l_bin add_zeros;
run;

title "Frequency distribution of mean lengths";
proc tabulate format=10.0 noseps formchar="             ";
var mean_length;
```

```
class gear mean_l_bin;
table mean_l_bin , gear*(mean_length*n) /misstext="0" rts=8;
run;


*==========================================================================
=======================================;
*==========================================================================
====================;
* Develop "average" length frequency distribution";

* Count number of surveys;


proc summary data=sum1;
by gear;
var survey;
id dummy;
output out=n_surveys n=n_surveys;
run;


proc sort data=sum1;
by gear survey;
run;

proc sort data=a1;
by gear survey;
run;


data freq1;
merge sum1 a1;
by gear survey;
run;

data freq2;
set freq1;
percent=number/sum_number*100;
run;

proc sort;by gear inch_group;run;

proc print;
run;


proc summary;
by gear inch_group;
id dummy;
var percent;
output out=freq3 sum=sum_percent;
run;


data freq4;
merge freq3 n_surveys;
by gear;
mean_percent=sum_percent/n_surveys;
run;
```

```
proc sort;
by gear inch_group;
run;

proc print;
run;


proc tabulate format=8.2 noseps formchar="                ";
class inch_group gear ;
var mean_percent;
table inch_group*(mean_percent*sum) all*mean_percent*sum , gear /
misstext="0";
run;
```

```
OPTIONS PS=250 ls=200 nocenter nodate;

libname h_drive "C:\Data\dnr\Resource Inventory Lake Reporting Template Aug
2005";

*==========================================================================
==============;
* Data selection area;
%let sp="YEP";      *Select 3-letter species code here;
%let min_month=5;  *Select starting month to include in analysis;
%let max_month=6;  *Select ending month to include in analysis;
*==========================================================================
==============;

* Note - this data set needs to be species specific and specific to time of
year;
* Note - need to program this in so it can do the merges by matching
species and age;
data state_average;
input species $ age standard_length begin_month end_month;
cards;
YEP 1 4.0 6 7
YEP 2 5.7 6 7
YEP 3 6.8 6 7
YEP 4 7.8 6 7
YEP 5 8.7 6 7
YEP 6 9.7 6 7
YEP 7 10.5 6 7
YEP 8 11.3 6 7
YEP 9 11.7 6 7
YEP 10 12.1 6 7
;
run;

* Note - be careful about input for species.  Issues include species codes
with more than 3 letters and with spaces.
Also, the location may shift from 42 to 44.;
data ages;
infile "C:\Data\dnr\Resource Inventory Lake Reporting Template Aug
2005\age_data.prn";
input survey month year species  $ 42-44 dummy $ 45-48 age_class $ length
inch_group;
if age_class ="0" then age=0;
if age_class ="I" then age=1;
if age_class ="II" then age=2;
if age_class ="III" then age=3;
if age_class ="IV" then age=4;
if age_class ="V" then age=5;
if age_class ="VI" then age=6;
if age_class ="VII" then age=7;
if age_class ="VIII" then age=8;
if age_class ="IX" then age=9;
if age_class ="X" then age=10;
if age_class ="XI" then age=11;
if age_class ="XII" then age=12;
if age_class ="XIII" then age=13;
if age_class ="XIV" then age=14;
if age_class ="XV" then age=15;
if age_class ="XVI" then age=16;
if age_class ="XVII" then age=17;
if age_class ="XVIII" then age=18;
if age_class ="XIX" then age=19;
```

```
if age_class ="XX" then age=20;

run;

*Select out target species;

data &sp;

set ages;
if species=&sp;
run;

proc sort data=&sp;
by survey;
run;

*proc print;
*run;

* First step is to figure out what months data are available from;

proc tabulate format=5. noseps formchar="              ";
class survey month;
var length;
table survey all, month*length*n all/rts=8;
run;


* Select out desired months;
data selected;
set &sp;
if month<&min_month then delete;
if month>&max_month then delete;
run;

data strat;
set h_drive.stratum;
run;

proc sort;
by survey;
run;

data all;
merge selected strat;
by survey;
if size_stratum="T" then stratum="Tiny "||depth_stratum;
if size_stratum="S" then stratum="Small "||depth_stratum;
if size_stratum="M" then stratum="Medium "||depth_stratum;
if size_stratum="L" then stratum="Large "||depth_stratum;

* This deletes out surveys without age data for the species of interest;
if month=. then delete;
run;

*==========================================================================
====================================;
* Begin data summary and analysis;
*==========================================================================
====================================;
title "Mean length at age by survey";
proc sort;
```

```
by survey age ;
run;

proc summary;
by survey age;
var length;
output out=mean_l_age mean=mean_l_age n=n stderr=se_l_age;
run;
proc print;
run;

proc tabulate noseps format=6.2;
class survey age;
var mean_l_age;
table survey , age*mean_l_age*mean/rts=8;
run;

* Compute overall mean length at age, ignoring missing data;
proc sort;by age;run;
proc summary;
by age;
var mean_l_age;
output out=mean_of_means mean=grand_mean_l_age n=n stderr=se_grand_mean;
run;

proc print;
run;

*==========================================================================
======;
* Compute growth indices, based on published state averages as well as
internal means;
data growth_standards (drop=_type_ _freq_ se_grand_mean n);
merge state_average mean_of_means;
by age;
if age=0 then delete;
run;

proc sort data=mean_l_age;by age;run;

data indices (drop=se_l_age);
merge growth_standards mean_l_age;
by age;
if age=0 then delete;
if n<5 then delete;
index1=mean_l_age - standard_length;
index2=mean_l_age - grand_mean_l_age;
run;

proc gplot;
plot standard_length*age grand_mean_l_age*age /overlay;
run;


proc sort;by survey;run;

proc summary;
by survey;
var index1 index2;
output out=indices1 mean=state_avg_dev internal_avg_dev;
run;
```

```
proc print;
run;

proc means;
var state_avg_dev internal_avg_dev;
run;

proc gplot;
plot state_avg_dev *internal_avg_dev;
run;
```