



Getting ready...

Formatting your data

Clearly, the first step in any analysis is gathering and collating your data. We'll assume that at the minimum, you have records for the individually marked individuals in your study, and from these records, can determine whether or not an individual was "seen" (either captured, or sighted) on a particular occasion.

- Most typically, your data will be stored in what we refer to as a "vertical file" - where each line in the file is a record of when a particular individual was seen.
- For example, consider the following table, consisting of some individually identifying mark (ring or tag number), and the year. Each line in the file (or, row in the matrix) corresponds to the animal being seen in a particular year.

Tag Number	Year
1147-38951	73
1147-38951	75
1147-38951	76
1147-38951	82
1147-45453	74
1147-45453	78

- However, while it is easy and efficient to record the observation histories of individually marked animals this way, the "vertical format" is not, unfortunately, at all useful for capture-mark-recapture analysis.

- As discussed at length in Lebreton et al. (1992), the preferred format for CMR data is the **capture-history**. The capture history is a contiguous series of "1" and "0", where "1" indicates that an animal was recaptured (or otherwise known to be alive and in the sampling area), and "0" indicates the animal was not recaptured (or otherwise seen).
- Take for example, the individual in the preceding table with tag number 1147-38951. Suppose that 1973 is the first year of the study, and that 1985 is the last year of the study. Examining the table, we see that this individual was captured and marked during the first year of the study, was seen periodically until 1982, when it was seen for the last time.
- The corresponding capture-history for this individual would be:

1011000001000

- In other words, the individual was seen in 1973 (the starting "1"), not seen in 1974 ("0"), seen in 1975 and 1976 ("11"), not seen for the next 5 years ("00000"), seen again in 1982 ("1"), and then not seen again ("000").
- While this is easy enough in principal, you surely don't want to have to construct capture-histories manually. Of course, this is precisely the sort of thing that computers are good for - large-scale data manipulation and formatting.
- Unfortunately, SURGE does not have a facility for data manipulation. Of the available software for mark-recapture analysis, only POPAN has a facility for data selection.
- Thus, in general, you'll have to write your own program to convert the typical "vertical" file into capture-histories. In fact, if you think about it a bit, you realize that in effect what you need to do is to take a vertical file, and "transpose" it into a horizontal file - where fields to the right of the individual tag number represent when an individual was recaptured or resighted.
- However, while the idea of a matrix transpose seems simple enough, there is one rather important thing that needs to be done - your program must insert the "0" value whenever an individual was not seen.
- We'll assume for the purposes of this book that you will have some facility to put your data into the proper capture-history format. Of

course, you could always do it by hand, if absolutely necessary!

- Once your data are formatted as capture histories, we are ready to introduce 2 different file formats: the RELEASE format, and the SURGE format.
- You may guess that the SURGE format is what program SURGE uses. You are, perhaps obviously, correct. So, you may wonder why we need to discuss 2 different formats (RELEASE and SURGE), instead of just SURGE alone.
- The reasons we consider both formats are two-fold. First, the initial step in any CMR analysis should be goodness of fit testing. Perhaps the most commonly used software for testing goodness of fit to the standard time-dependent models (discussed in considerable detail starting in Chapter 3) is program RELEASE.
- So, since running program RELEASE for GOF testing is our first step (see the Appendix for GOF testing using RELEASE), we might as well store our data in RELEASE format in the first place. The SURGE format does not allow GOF testing, since (as you will soon see), the SURGE format is a data summary which does not contain sufficient information for GOF testing.
- Fortunately, the RELEASE format is very intuitive - the actual data are stored as capture-histories, which provides an obvious and intuitive link to the CMR analyses.
- Of course, the RELEASE formatted file contains several other pieces of information, specifically, the control statements which determine how program RELEASE interprets the data, and other details concerning the GOF testing, but at its core, the RELEASE data file contains capture histories.
- Once you've gone to the effort of correctly formatting your RELEASE file, you can then convert it into a SURGE formatted file, using a utility called RELTOSUR.EXE. The RELTOSUR utility, available via the WWW (see <http://www.biol.sfu.ca/cm/surge>), is small, very fast, and turns your capture histories into files SURGE can use without causing you any major problems.
- Of course, it would be useful if RELTOSUR was built into SURGE itself, rather than as a separate utility. We suspect that this will be implemented in future releases of SURGE (no pun intended!).
- So, why the need to have a proprietary SURGE format? Well, as we

will see shortly, the SURGE format condenses a lot of the information “stored” in capture-histories into a summary data matrix which is much more efficiently analyzed than the capture-histories alone. It is in part the use of the SURGE format which allows SURGE to find solutions to model fits very quickly, often many orders of magnitude faster than other software. So, in a sense, the inconvenience of having to deal with 2 different file formats is balanced by the significant advantages in execution speed for program SURGE. However, one thing that the “summary format” precludes is the use of individual covariates (i.e., relating survival or recapture probabilities of individuals to individual characteristics, like body mass, for example).

- We will now go through the basic details of both the RELEASE and SURGE formats.

RELEASE data format

- The definitive documentation for program RELEASE is the Fisheries Monograph (5) by Burnham et al. (1987), published by the American Fisheries Society (sometimes known in the trade as the “big blue book”, for its size and blue cover- the full citation for this reference is noted in the appendix). The first 386 out of 411 pages of the book cover the theory and application of tag-recovery studies, primarily in a fisheries context. It is an excellent review of the basic ideas of CMR, as well as a good primer for the statistical theory of goodness of fit and power testing. It is not always easy reading, but it is worth the effort. The remaining 26 pages constitutes the only published documentation for program RELEASE that we are aware of, and are not an easy read.
- Program RELEASE has a large number of options, including an excellent facility to simulate data sets under a variety of model structures.
- For the purposes of this book, we'll concentrate only on the absolute minimum number of elements of the RELEASE format necessary to create a RELEASE formatted file that the utility RELTOSUR will properly convert.
- The minimum RELEASE data file contains 4 elements: (1) a title statement, (2) a data input formatting statement, (3) the capture histories, and (4) a stop statement. Again, all we're going to do in this

book is show you what the minimum RELEASE file looks like. For more detailed explanation of these 4 elements, especially the data input formatting statement, we'll leave it to you to track down a copy of the "blue book", and study the documentation.

- Here is a typical RELEASE file:

```

PROC TITLE HERE IS A TITLE;
PROC CHMATRIX OCCASIONS=9 GROUPS=1;
110100111    23;
110000101    4;
101100011    1;
010010111   13;
010000101    4;
001011011   12;

      .
      {RECORDS DELETED}
      .
000000010    45;
000000011    4;
PROC STOP;

```

- The first line is the PROC TITLE statement. You simply type the words PROC TITLE, and then whatever is written to the right of PROC TITLE (in this case, HERE IS A TITLE) will be interpreted by RELEASE as the title. The utility RELTOSUR ignores it completely, but expects it to be in the RELEASE file. RELEASE files use PROC statements, much as SAS does.
- The second line is the data input formatting, or CHMATRIX line. If you look at it closely, it should be fairly self-explanatory. First the PROC CHMATRIX part tells RELEASE that the data are stored in capture-history format (hence the CHMATRIX, for capture-history matrix). Then, the number of occasions - in this case, 9 occasions. Note that there is nothing which indicates the actual chronological date of

the various occasions. Neither RELEASE (nor SURGE) care when the data were collected. Then, the GROUPS=1 statement. In this study, there is only one group of individuals. If we were interested in comparing multiple groups, for examples males versus females, the number of groups in this line would change accordingly. The RELTOSUR utility "looks" at these values, and uses them in the conversion process. More on GROUPS in a moment.

- Next, the capture histories themselves. In this case, each capture history is followed by a number. This number is the frequency of all individuals having a particular capture history. This is not required (you could have a capture history for each individual in the data set), but is often more convenient for large data sets. For example, the summary capture history

```
110000101    4;
```

could also be entered in the RELEASE file as:

```

110000101    1;
110000101    1;
110000101    1;
110000101    1;

```

- Note that each line in the file ends in a semi-colon, including the capture-histories. The control or command lines ending in semi-colons should be familiar to anyone who uses SAS. However, unlike SAS, data lines are also followed by a semi-colon.
- The final statement, PROC STOP, merely signifies to both RELEASE and RELTOSUR that the end of the capture-histories has been reached.
- So, at minimum, you need a PROC TITLE statement, a PROC CHMATRIX OCCASIONS=*nnn* GROUPS=*yyy* statement, the capture histories, and a PROC STOP statement.
- How would you change the RELEASE file if you had more than one group? Suppose we wanted to compare males and females of some species?

- In fact, it is easy to format multiple groups for RELEASE or RELTOSUR. You simply change the GROUPS=1 part of the PROC CHMATRIX statement to GROUPS=2, and then add a second column of frequencies to the capture histories for the other sex. For example, our example RELEASE file might look like:

```

PROC TITLE HERE IS A TITLE;
PROC CHMATRIX OCCASIONS=9 GROUPS=2;
110100111    23   17;
110000101     4    2;
101100011     1    3;
010010111    13    7;
010000101     4    0;
001011011    12    4;
.
{RECORDS DELETED}
.
000000010    45   49;
000000011     4    3;
PROC STOP;
    
```

- If you are using individual records, rather than summary frequencies (see preceding page), you need to do something a little less obvious - you will have to use a 0/1 “dummy” variable coding scheme to code group associations (in this case, male or female). Dummy variable coding is presented in considerable detail in Chapter 6, in a very different context.
- In this example, the summary capture history

110000101 4 2;

would become

```

110000101    1    0;
110000101    1    0;
110000101    1    0;
110000101    1    0;
110000101    0    1;
110000101    0    1;
    
```

- Now that we have our RELEASE file, we’ll convert it to SURGE format, using RELTOSUR.

Removing animals

- Occasionally, you may choose to remove animals from the data set at a particular occasion. For example, your experiment may require you to remove an animal after its first recapture, or because it is injured, and so forth.
- The standard capture history we have described so far records presence or absence only. How do we “tell” RELEASE that we are removing one or more animals from the study? In fact, it is very easy, all you do is change the sign on the number in the “frequency column” from positive to negative. Negative frequencies indicates the animal (or animals) with a particular capture history were removed from the study.
- Consider the following example (a piece of a larger RELEASE file).

```

100100    1500    1678;
100100     -23     -25;
100011     -18     -24;
100100    1500    1678;
100100     -81     -57;
    
```

- In this example, we see that we have 2 groups, and 6 occasions. In the first line in the table, we see that there were 1500 and 1678 individuals in both groups marked on the first occasion, not seen on the next 2 occasions, seen on the fourth occasion, and not seen again.
- In the second line, we see exactly the same capture history, but with the numbers “-23” and “-25”. What the negative values mean is that 23 and 25 individuals in both groups were marked on the first occasion, not seen on the next two occasions, were seen on the fourth occasion and were removed from the study (clearly, if they were removed, they cannot have been seen again).

Running the RELTOSUR utility

- Let’s assume that the name of our RELEASE file is SAMPLE.REL. In general, we prefer using file extensions (the last 3 characters in the DOS 8.3 file naming convention) to reflect the type of file you’re working with. Neither RELEASE, RELTOSUR or SURGE “care” what you call your files. However, we have found it useful to use a common “rule” for file extensions - it is a simple way to help you keep track of what files are what (.REL for RELEASE, .SUR for SURGE). We will make this suggestion periodically throughout this book.
- Once you have the SAMPLE.REL file ready to go, you need to invoke the RELTOSUR utility. It is probably simplest to put the utility (the RELTOSUR.EXE file) somewhere in the PATH.
- First, check to see that the directory containing the RELTOSUR executable is in the PATH. You can do this by simply typing

PATH <enter>

If the directory containing RELTOSUR is not in the PATH, then edit the AUTOEXEC.BAT file to add the appropriate directory to the PATH, and then re-boot your computer to make this change effective. If RELTOSUR is in the PATH, then simply switch to the directory into which you placed the file SAMPLE.REL), and start the program by typing:

RELTOSUR <enter>

- if the RELTOSUR program is working properly, it will present you with the opening screen (Fig. 2.1).

```

CONVERT RELEASE FORMAT TO SURGE4 FORMAT
INPUT FILE <RECAPTURE HISTORIES, RELEASE FORMAT> ? _

```

Fig. 2.1

- If nothing happens (or you get some typically obscure DOS error message), this is a good indication that either (a) the directory containing the RELTOSUR program is not in the PATH (go back and check that it is), or (b) RELTOSUR has been corrupted.
- Once RELTOSUR is running, simply type in the name of the file you want to convert (in this case, SAMPLE.REL), and hit the <enter> key.
- In this example, we have 9 occasions, and 1 group - we have “told” RELTOSUR this on the PROC CHMATRIX line in the SAMPLE.REL file (see preceding pages).
- RELTOSUR confirms this for us by printing this information, as well as the total number of different capture histories, onto the screen (Fig. 2.2).

```

CONVERT RELEASE FORMAT TO SURGE4 FORMAT
INPUT FILE <RECAPTURE HISTORIES, RELEASE FORMAT> ? example.rel
200      RECAPTURE HISTORIES
1        GROUPS
9        OCCASIONS
OUTPUT FILES <RECAPTURE HISTORIES, SURGE FORMAT>
GROUP 1  ? example.sur

```

Fig. 2.2

- RELTOSUR will then prompt you for the name you want to give to the

SURGE formatted file (Fig. 2.2 - bottom). In this case, call the file SAMPLE.SUR (with the .SUR extension to indicate SURGE format), and hit the <enter> key.

- Since there is only 1 group in SAMPLE.REL, RELTOSUR will perform the conversion, and then exit. If there were more than one group, RELTOSUR would prompt you for the names of the files you wanted for the other groups - one prompt for each group.
- Now that you've created SAMPLE.SUR, you could immediately jump ahead and start analyzing your data using SURGE (Chapter 3-10).
- However, it is worth taking a few moments to examine the format of the SURGE file, just to get an idea of what is going on. It is always worth doing a bit of "exploration", rather than just blindly accepting what the computer has done (description of the SURGE format is also presented in the documentation which is generally supplied with SURGE).

The SURGE file format

- Each SURGE format file contains the following. The first row contains the number of recapture occasions ($K-1$, where K is the total number of occasions).
- The second row contains the number of **newly** marked individuals in each cohort (i.e., the number of newly marked individuals released at each occasion. Let's represent these releases of newly marked individuals as b_i , for $i=1$ to $K-1$ occasions).
- Then, beginning with the third row, the number of individuals marked at occasion i subsequently recaptured at time j , where $j>i$. Let these numbers be represented as a_{ij} , for $i=1$ to $K-1, j=2$ to K - in other words, the numbers caught at time j among the b_i individuals newly marked and released at occasion i .
- Let's examine what the .SUR file would look like so far. Consider $K=4$ occasions. Thus, the .SUR file would have the following structure:

```

3
b1 b2 b3
a12 a13 a14
a23 a24
a34

```

- Is this all there is to a SURGE formatted file? Well, if you think about it, you'd realize we're missing one very important piece of information - the number of individuals seen for the last time on a particular occasion. Let c_{ij} - for $i=1$ to $K-1, j=1$ to $K-1$ be the number of individuals seen for the last time at occasion j among those marked at occasion i . We note that these c_{ij} values also include those not released (i.e., removed from the study), but we'll ignore that complication for the moment, assuming that all animals, if alive, are released back into the population following recapture.
- If we add the c_{ij} values to our example where $K=4$ occasions, our .SUR file would look like

```

3
b1 b2 b3
a12 a13 a14
a23 a24
a34
c11 c12 c13
c22 c23
c33

```

- This is the complete .SUR formatted file, unless you have removed marked animals following recapture (either intentionally, or due to capture mortality). As you can see, it is very much a condensed version

of the information stored in capture-histories. The capture-histories contain all of the information needed for any analysis, whereas the SURGE format is a synthesis of this information. Thus, while you can convert a RELEASE file into a SURGE file (REL → SUR), you can't go the other direction.

- Therefore, for a variety of reasons, we strongly suggest first collating your data into capture-histories, in a RELEASE formatted file.

That's it! You're now ready to learn how to use SURGE. Before you leap into the first "SURGE" chapter (Chapter 3), take some time to consider that SURGE will always do its "best" to analyze the data you feed into it. However, it assumes that you will have taken the time to make sure your data are correct. If not, you'll be the unwitting victim to perhaps the most telling comment in data analysis: "garbage in...garbage out". Take some time at this stage to make sure you are confident in how to properly create and format your files. Time spent now will save you considerable aggravation later, when you realize (typically after you're "finished") your data weren't correct in the first place! Of course, we've never experienced this ourselves! Well, maybe once or twice...

