



Appendix - GOF Testing with Program *RELEASE*

In this Appendix we will provide a **brief** introduction to a very important topic - Goodness of Fit testing (GOF). All of the models we have discussed in this book make very specific assumptions that must be tested before using *SURGE* - thus, as a first step, you need to confirm that your starting model adequately fits the data, using GOF tests. You'll note that we mentioned this, directly or indirectly, at several points throughout the book, without specifically telling you "how". Much of this material is also presented in considerable detail in Lebreton *et al.* (1992).

- There are a number of ways in which GOF testing can be accomplished, and a variety of GOF procedures have been implemented in several different CMR software applications. For example, programs *RELEASE*, *SURVIV*, *JOLLY*, and *JOLLYAGE* all provide GOF statistics for various models. However, *SURGE* does NOT provide any 'built-in' GOF testing.
- We will concentrate on the GOF statistics generated by program *RELEASE*, written by Gary White (*RELEASE* can be obtained via the World Wide Web - see p. 9-5). We note that although *RELEASE* also does "survival analysis" - in the CMR context, many of the survival tests generated by *RELEASE* are specialized models more appropriate for specific kinds of fisheries tag-release experiments, and may not be generally useful for other types of studies. These "survival" tests, the GOF tests, and the complete documentation for program *RELEASE*, are presented in considerable detail in:

Burnham, K.P., Anderson, D.R., White, G.C., Brownie, C., & Pollock, K.H. (1987) Design and analysis methods for fish survival experiments based on release-recapture. *American Fisheries Society Monograph 5*. American Fisheries Society (5410 Grosvenor Lane, Bethesda, Maryland 20814).

- This is the "blue book" mentioned in Chapter 2.
- Although the GOF tests generated by *RELEASE* are described in detail in the "blue book", we also suggest that the reader consult

Pollock, K.H, Nichols, J.D., Brownie, C. & Hines, J.E.
(1990) Statistical inference for capture-recapture experiments. *Wildlife Monographs, 107*, 1-97

- This monograph is one of the standard "primers" on CMR analysis (for both closed and open models - the Lebreton *et al.* monograph and the "blue book" are both devoted exclusively to open models), and is an excellent reference on the general question of GOF testing for some of the standard open models - see especially pp. 22-24. The Pollock *et al.* monograph in fact introduces programs *JOLLY* and *JOLLYAGE*, and goes into some detail on the way in which the GOF statistics for both programs are constructed. In some cases, the GOF statistics provided by *JOLLY* and *JOLLYAGE* are identical to those provided in *RELEASE*.
- In this appendix, we will introduce briefly how to use *RELEASE* to generate GOF statistics, and give some broad suggestions for how to interpret lack-of-fit (from both a statistical and biological point of view), and what remedies are available. In writing this, we've assumed that you've either gone through the rest of this book, or are already familiar with the basics of CMR analysis and model fitting.

Program Release - TEST 2 & TEST 3

- Program *RELEASE* generates 3 standard "tests", which are given the absolutely uninformative names "TEST 1", "TEST 2", and "TEST 3". The latter 2 tests, **TEST 2** and **TEST 3**, together provide the GOF statistics for the reference model (the time-dependent CJS model). TEST 1, mentioned briefly in Chapter 5, is an omnibus test that is generated ONLY if you are comparing groups, and tests the following hypothesis:

H_0 : all parameters ϕ_i and p_i have the same value across treatment groups (i.e., there is no difference in survival (ϕ_i) or recapture (p_i) considered simultaneously among groups).

H_A : at least some values for either ϕ_i or p_i (or both) differ among groups.

- The big advantage of using SURGE or one of the other applications available for CMR analysis, is that you can separately model differences in either survival or recapture rate independently. TEST 1 does not do this - it only tests for an “overall” difference among groups. Since this severely limits its utility, we will not discuss use of TEST 1 - in fact, we actively discourage its use, since it is possible to do far more sophisticated analyses if you have capture histories from individually marked animals (although TEST 1 may still be of use when complete capture histories are not available - see the “blue book” for use of RELEASE and TEST 1 under alternative capture protocols).
- While TEST 1 may be of limited use, TEST 2 and TEST 3 together are quite useful for testing the GOF of the standard time-dependent CJS (Cormack-Jolly-Seber) model to the data (this model was first presented in detail in Chapter 4).
- What do we mean by “lack of fit”? Specifically, we mean that the arrangement of the data do not meet the expectations determined by the assumptions underlying the model. These assumptions, sometimes known as the CJS assumptions are:
 1. *Every marked animal present in the population at time (i) has the same probability of recapture (p_i)*
 2. *Every **marked** animal in the population immediately after time (i) has the same probability of surviving to time ($i+1$).*
 3. *Marks are not lost or missed.*
 4. *All samples are instantaneous, relative to the interval between occasion (i) and ($i+1$), and each release is made immediately after the sample.*
- For now, we will assume that assumptions 3 and 4 are met. It is assumptions 1 and 2 which are typically the most important in terms of GOF testing. In fact, TEST 2 and TEST 3 in RELEASE, as well as the GOF tests in other software, directly test for violations of these two assumptions (in one form or another).
- Let’s expand somewhat on assumptions 1 and 2.
- Assumption 1 says that all marked animals in the population have the same chances of being captured at any time (i). What would be the basis for violating this assumption? Well, suppose that animals of a particular age or size are more (or less) likely to be captured than animals of a different age or size? Or, suppose that animals which go through the process of being captured at occasion (i) are more (or less) likely to be captured on a later occasion than animals who were marked at some other occasion? For estimation of survival in open populations, marked animals have the same probability of recapture. For estimation of population size (abundance), both marked *and* unmarked animals must have the same probability of capture.
- What about assumption 2? Assumption 2 says that, among the marked individuals in the population, all animals have the same probability of surviving, regardless of when they were marked. In other words, **animals marked at occasion ($i-1$) have the same probability of surviving from (i) to ($i+1$) as do animals marked on occasion (i).** When might this assumption be violated? One possibility is that individuals caught early in a study are more (or less) prone to mortality than individuals caught later in the study. Or, perhaps you are marking young individuals. An individual captured and marked as an offspring at ($i-1$) will be older, or larger, or possibly of a different breeding status, at occasion (i), while offspring marked at occasion (i) are just that, offspring. As such, the offspring marked at ($i-1$) may show different survival from (i) to ($i+1$) than offspring marked at (i), since the former individuals are older, or larger, or somehow “different” from individuals marked at (i).
- For both TEST2 and TEST3 we have noted several reasons why either TEST 2 or TEST 3 might be violated. The examples noted are by no means an all-inclusive list - there are **many** other ways in which either or both tests could be violated. While violation of the underlying model assumptions has a specific statistical consequence (which we will deal

with shortly), it may also serve to point out something interesting biologically. For example, suppose all animals are not equally likely to be captured at any occasion - why? Does this reveal something interesting about the biology?

- We'll approach GOF testing in 2 steps. First, we'll describe the "mechanics" of how to run RELEASE to generate the TEST 2 and TEST 3 results. Then, we'll discuss the mechanics of how these two tests are constructed, and how to interpret them.

Running RELEASE

- Earlier in this chapter we discussed the minimum RELEASE formatted file needed to run the "RELEASE to SURGE" conversion utility, RELTOSUR. In fact, it is also the minimum file for running program RELEASE. The "important" line in the RELEASE file is the PROC CHMATRIX statement. In Chapter 2, we presented an example of a RELEASE file containing capture history data for 1 group, and 9 occasions. The PROC CHMATRIX line was written as:

```
PROC CHMATRIX OCCASIONS=9 GROUPS=1;
```

- The PROC CHMATRIX statement must include (at least) the GROUPS and OCCASIONS statements. However, there are a number of other options which can also be applied to this procedure. One of these options is DETAIL - as its name implies, the DETAIL option provides "detailed" information about something. The "something" is in fact detailed information concerning TEST 2 and TEST 3. When the DETAIL option is in effect, RELEASE provides the individual contingency tables (including observed and expected frequencies) upon which they are based (discussed below), as well as the summary statistics for all batches pooled. If you have a data set with a large number of occasions, this can generate a very large amount of output.
- The opposite to the DETAIL option is the SUMMARY option, which forces RELEASE to print only the summary TEST 2 and TEST 3 results for each batch separately and all batches pooled.
- The way you choose either the DETAIL or SUMMARY option

depends on the version of RELEASE you are using (unfortunately, there are a number of different versions of RELEASE in circulation). In some versions, DETAIL is the default, and you need not make any changes to the PROC CHMATRIX statement on the preceding page - the "detailed" TEST 2 and TEST 3 tabulations will be generated by default. However, in other versions of RELEASE, you must explicitly add the word DETAIL to the PROC CHMATRIX statement:

```
PROC CHMATRIX OCCASIONS=9 GROUPS=1 DETAIL;
```

- However, adding the DETAIL option explicitly will always work - if your version of RELEASE doesn't require it, the DETAIL option will simply be ignored.
- To use the SUMMARY option (instead of DETAIL), you would type

```
PROC CHMATRIX OCCASIONS=9 GROUPS=1 SUMMARY;
```

- Which option should you use, DETAIL or SUMMARY? We recommend using the DETAIL default, if only because there are instances where the summary statistics are insufficient to give you a clear indication as to what is going on in your data.
- How do you run RELEASE? There are two ways you can run RELEASE - interactively, or in batch mode. The interactive mode will allow you to build the RELEASE file in a step-by-step fashion. However, we've already covered the basics of "manually" constructing the RELEASE file in Chapter 2 - so we can go ahead and run RELEASE in batch mode. To do this is very straightforward - simply type:

```
RELEASE I=<INPUT FILE> O=<OUTPUT FILE> <enter>
```

- If our RELEASE file is called TEST.REL, and we want our results to be written to a file called TEST.LST, then we would type:

```
RELEASE I=TEST.REL O=TEST.LST <enter>
```

- That's it! After RELEASE is finished, all you'll need to do is look at the output file (TEST.LST in this example) with your favourite editor.
- At the top of this output file there will be some information concerning recent updates to the RELEASE program, and some statement concerning limits to the program (maximum number of groups, or occasions). Then, depending on the version of RELEASE you are using, you will see a listing of the individual capture histories in your data set, plus a summary tabulation of these histories known as the **reduced m -array**. The m -array contains summary information concerning numbers of individuals released at each occasion, and when (and how many) of them were captured at subsequent occasions. The reduced m -array will be discussed in more detail later.
- These m -array tabulations are then followed by the TEST 3 and TEST 2 results for each batch (respectively), followed in turn by the summary statistics.

TEST 2

- TEST 2 deals only with those animals known to be alive between (i) and $(i+1)$. This means we need individuals marked at or before occasion (i) , and individuals captured at or later than $(i+1)$. If they were alive at (i) , and captured at or later than $(i+1)$, then they must have been alive in the interval from occasion (i) to $(i+1)$.
- From these individuals, we can construct the general form of the contingency table referred to as TEST2.

	seen at $(i+1)$	seen after $(i+1)$
not seen at (i)	f	f
seen at (i)	f	f

- In other words, “is the probability of being seen at occasion $(i+1)$ a function of whether or not you were seen at occasion (i) , given that you survived from (i) to $(i+1)$?”. Under assumption 1, all marked animals should be equally catchable (or visible) at occasion $(i+1)$ independent of whether or not they were captured at occasion (i) .
- In some recent versions of RELEASE, TEST2 is subdivided into two component tests: TEST2.Ct and TEST2.Cm (see Pradel, R. (1993) Flexibility in survival analysis from recapture data: handling trap-dependence. In Marked Individuals in the Study of Bird Populations (ed. J-D. Lebreton & P. North). Birkhauser-Verlag, pp 29-37). We believe this extension of the standard “TEST2” is a good and useful one, and will focus our discussion on the assumption your version of RELEASE provides these statistics. From these individuals, we can construct the first of two “types” of contingency tables which together comprise TEST 2.
- This test depicted in the table in the left-hand column corresponds to TEST 2.Ct,
- The other “part” of TEST 2 is TEST2.Cm. TEST2.Cm is a simple extension of TEST2.Ct. Of those individuals surviving from (i) to $(i+1)$, some were seen at $(i+1)$, while some were seen after $(i+1)$. Of those not seen at $(i+1)$, but seen later, does “when” they were seen differ as a function of whether or not they were captured at occasion (i) ? In other words:

	when seen again?					
	$(i+2)$	$(i+3)$	$(i+4)$	$(i+5)$...	$(i+n)$
not seen at (i)	f	f	f	f	f	f
seen at (i)	f	f	f	f	f	f

- So, TEST2.Cm asks “of those animals not seen at $(i+1)$, but known to be alive at $(i+1)$, does when they were next seen $(i+2, i+3...$) depend on

whether or not they were seen at (i)?”. Again, we see that TEST2.Cm deals with capture heterogeneity.

- TEST 2 (in general) is sensitive to short-term capture effects, or non-random temporary emigration. It highlights failure of the homogeneity assumption (assumption 1), among animals and between occasions.
- In practice, TEST 2 is perhaps most useful for testing the basic assumption of “equal catchability” of marked animals. In other words, we might loosely refer to TEST 2 as the “recapture test”.

TEST 3

- In general, TEST 3 tests the assumption that all marked animals alive at (i) have the same probability of surviving to ($i+1$) - the second assumption (p. 2-7).
- TEST 3 asks: “of those individuals seen at occasion (i), how many were ever seen again, and when?”. As with TEST 2, some of the individuals seen at occasion (i) were seen for the first time at that occasion, while others had been previously seen (marked). Does whether or not they were ever seen again depend on this conditioning? The first part of TEST 3, known as TEST3.SR, is shown in the following contingency table:

seen at (i)	seen again	not seen again
seen before	f	f
not seen before	f	f

- In other words, does the probability that an individual known to be alive at occasion (i) is ever seen again depend on whether it was marked at or before occasion (i)? TEST 3.SR bears a clear structural similarity to TEST 2.Ct.

- Analogous to TEST2.Cm, there is also a TEST3.Sm, which asks “...among those animals seen again, does WHEN they were seen depend on whether they were marked on or before occasion (i)?”. TEST3.Sm is depicted in the following contingency table:

	when seen again?					
seen at (i)	($i+1$)	($i+2$)	($i+3$)	($i+4$)	...	($i+n$)
seen before	f	f	f	f	f	f
not seen before	f	f	f	f	f	f

- So, in a very loose sense, TEST 2 deals with “recapture problems”, while TEST 3 deals with “survival problems” (although there is no formal reason to make this distinction - it is motivated by our practical experience using RELEASE).
- If you think about it, these tables should make some intuitive sense: if assumptions 1 and 2 are met, then there should be no difference among individuals if or when they were next seen conditional on whether or not they were seen on or before occasion (i).
- Lets consider a simple example of GOF testing with RELEASE. We created a simulated data set which we constructed so that it “should” meet CJS assumptions. Does RELEASE confirm our expectations?
- The simulated data set had 6 occasions, and 1 group. Both survival and recapture rates were allowed to vary with time (in the range 0.5 to 0.7 for both parameters). We used a staggered release protocol, with new individuals marked and released at each occasion. We arbitrarily used 150 as a constant number of new individuals released.
- We ran RELEASE using the DETAIL option (discussed earlier), so we could look at the TEST 2 and TEST 3 results on a occasion-by-occasion basis.
- First, lets look at the reduced m -array table RELEASE generates as the

default (the other m -array presentation you can generate with RELEASE is the **full m -array** - this will be discussed later). Examination of the m -array will give you some idea as to “where the numbers come from” in the TEST 2 and TEST 3 contingency tables. However, as we’ll see, the reduced m -array alone is not always sufficient.

Observed Recaptures for Group 1 Control Group								
i	R(i)	m(i,j)					r(i)	
		j=	2	3	4	5	6	
1	150		41	9	4	0	0	54
2	191			64	22	2	2	90
3	223				69	15	8	92
4	245					71	15	86
5	238						72	72
m(j)			41	73	95	88	97	
z(j)			13	30	27	25	0	
Sums for the above Groups								
m.	0	41	73	95	88	97		
z.	0	13	30	27	25	0		
R.	150	191	223	245	238			
r.	54	90	92	86	72			

- The major items of interest in the m -array are the R_i , m_{ij} and r_i values.
- The R_i values are the number of individuals in total released on each occasion. For example, $R_1 = 150$ equals 150 individuals released on the first occasion - all newly marked. At the second occasion (R_2), we released a total of 191 individuals - 150 newly marked, and 41 individuals that had been marked at occasion 1 and been recaptured at occasion 2.
- The m_{ij} values are the number of individuals from a given release event which are *captured for the first time* at a particular occasion. For example, $m_{1,2} = 41$. In other words, 41 of the original 150 individuals marked and released at occasion 1 (i.e., R_1) were recaptured for the first time at occasion 2. At the third occasion ($m_{1,3}$), 9 individuals

marked and released at occasion 1 were recaptured for the first time, and so on.

- The r_i values are the total number of individuals captured from a given *release batch* (see below). For example, from the original $R_1 = 150$ individuals, a total of 54 were recaptured over the next 5 capture occasions. Neither the m_{ij} or r_i values distinguish between newly marked or re-released individuals - they are simply subtotals of all the individuals released at a given occasion. As we’ll see shortly, this limits the usefulness of the reduced m -array.



What do we mean by “release batch” (above)? Following “the blue book”, a *cohort* is a group of animals released at the same occasion - whether newly marked or not. However, as you’ll recall from Chapter 7 & 8, when using SURGE, we refer to a cohort as all animals marked at the same occasions. In this context, an animal does not change cohorts - it is a “fixed” characteristic of each animal. In the RELEASE context, cohort changes with each capture occasion. To prevent confusion, we use the term release batch, or simply batch, to refer to all animals (marked and unmarked) released on a given occasion.

- Following the reduced m -array are the results for TEST 3. Since there are 5 recapture occasions, there are as many as 7 total TEST 3 tables (4 for TEST3.SR and 3 for TEST3.Sm). Let’s consider just one of these tables - the first TEST3.SR table, for individuals captured on occasion 2. Why is this the first table? Well, recall what TEST3.SR compares - seen before versus not seen before - obviously, at occasion 1, NO animals were seen before. Thus, we start at occasion 2.
- Note that each contingency table (see following page) starts with a “loose” restatement of what is being tabulated - in this case “goodness of fit test of seen before versus not seen before against seen again versus not seen again by capture occasions”. You will also see comments concerning which group is being tested, and possibly something concerning “control group. By default, if you’re working with only one group, RELEASE assumes that it is a “control group” in a “control vs. treatments” experiments. Then, comes the contingency table itself.

Goodness of fit test of seen before versus not seen before against seen again versus not seen again by capture occasions.

Test for Group 1
Control Group
TEST 3.SR2: Animals captured on occasion 2

O:	23	18	41
E:	19.3	21.7	
C:	0.7	0.6	

O:	67	83	150
E:	70.7	79.3	
C:	0.2	0.2	

90 101 191
Chi-square=1.6885 (df=1) P=0.1938

- First, note the labeling: TEST3.SR2. The “TEST.3SR” part is obvious, the “2” simply refers to the second occasion (so, TEST3.SR3 for the third occasion, TEST3.SR4 for the fourth occasion, and so on). At occasion 2, a total of 191 individuals were released. Of these, 41 had been seen before, and 150 were newly marked individuals. In the first row of the contingency table, we see that of the 41 individuals seen before, a total of 23 (or 56%) of these individuals were ever seen again. In the second row of the table, we see that of the 150 newly marked individuals, a total of 67 (or 45%) were ever seen again. Where did the numbers 23 and 47 come from? Can we tell from the reduced *m*-array? Unfortunately, the answer is no. Why? Because the reduced *m*-array does not “keep track” of the fates of individuals depending on when they were marked. For this, you need a different type of *m*-array, known as the **full *m*-array**. To generate the full *m*-array, you need to modify the PROC CHMATRIX statement in the RELEASE file, by adding the statement FULLM. For our example, with 6 occasions and one group, we would write:

```
PROC CHMATRIX OCCASIONS=6 GROUPS=1 DETAIL FULLM;
```

- The full *m*-array for our data set is shown in the following column. As you can readily see, the full *m*-array contains much more information than the reduced *m*-array. In fact, it contains the entire data set!! If you have the full *m*-array, you have all the information you need to run a

CMR analysis. If you look closely at the full *m*-array, you’ll see why.

Release i	1	2	3	4	5	6	r(i)	R(i)- r(i)	
1	<1>	150	41< 0>	9< 0>	4< 0>	0< 0>	0< 0>	54	96
2		<01>	150	47< 0>	16< 0>	2< 0>	2< 0>	67	83
2		<11>	41	17< 0>	6< 0>	0< 0>	0< 0>	23	18
3			<001>	150	54< 0>	11< 0>	6< 0>	71	79
3			<101>	9	1< 0>	1< 0>	0< 0>	2	7
3			<011>	47	13< 0>	3< 0>	2< 0>	18	29
3			<111>	17	1< 0>	0< 0>	0< 0>	1	16
4				<0001>	150	44< 0>	8< 0>	52	98
4				<1001>	4	1< 0>	0< 0>	1	3
4				<0101>	16	5< 0>	2< 0>	7	9
4				<1101>	6	4< 0>	0< 0>	4	2
4				<0011>	54	12< 0>	5< 0>	17	37
4				<1011>	1	1< 0>	0< 0>	1	0
4				<0111>	13	3< 0>	0< 0>	3	10
4				<1111>	1	1< 0>	0< 0>	1	0
5					<00001>	150	45< 0>	45	105
5					<01001>	2	0< 0>	0	2
5					<00101>	11	3< 0>	3	8
5					<10101>	1	0< 0>	0	1
5					<01101>	3	0< 0>	0	3
5					<00011>	44	16< 0>	16	28
5					<10011>	1	1< 0>	1	0
5					<01011>	5	0< 0>	0	5
5					<11011>	4	1< 0>	1	3
5					<00111>	12	4< 0>	4	8
5					<10111>	1	0< 0>	0	1
5					<01111>	3	2< 0>	2	1
5					<11111>	1	0< 0>	0	1

- Let’s concentrate on the information needed to generate TEST3.SR2. From the preceding page, recall that of the 41 individuals marked at occasion 1, captured and re-released at occasion 2, 23 were seen again. What would the capture history of these 41 individuals be? - obviously “11” - marked at the first occasion, and recaptured at the second occasion. The “11” capture history is represented as {11} in the full *m*-array. Find this capture history in the 3rd line. To the right, you will see the number 41, indicating that there were 41 such individuals. To the right of this value are the totals, by capture occasion, of individuals from this group of 41 ever seen again. For example, 17 of this 41 were seen again for the first time at occasion 3, 6 were seen for the first time at occasion 4, and so on. In total, of the 41 {11} individuals released,

a total of 23 were seen again.

- You should now be able to see where the values in the TEST3.SR2 table came from.
- Now, consider the “statistical results”. Although the proportions seen again appear to differ between the two groups (56% for previously marked vs 45% for the newly marked), they are not statistically different ($\chi^2=1.689$ (df=1), P=0.194).
- What are the other 2 numbers in each of the cells? Well, if you look down the left side of the table you’ll get a hint - note the 3 letters “O”, “E” and “C”. “O” = the observed frequencies, “E” = the expected frequencies (under the null hypothesis of the test), and “C” represents the contribution to the overall table χ^2 value (summing the “C” values for all four cells yield 1.689. The “C” values are simply $(O-E)^2/E$).
- So, for individuals released at the second occasion, there is no significant difference in “survival” between newly marked and previously marked individuals.
- Following the last table (TEST3.SR5 - individuals released at occasion 5), RELEASE prints a simple cumulative result for TEST3.SR - which is simply the sum of the individual χ^2 values for each of the individual TEST3.SRn results.

**Cumulative result of TEST 3.SR over occasions for Group 1
Chi-square=8.7170 (df=4) P=0.0686**

- In the simulated data set, we see that TEST3.SR was not significant, although the significance was close at $\alpha=0.05$ - $\chi^2=8.72$ (df=4), P=0.069. Examination of the individual TEST3.SRn tables showed that most of the “near significance” of the cumulative result came from one table (TEST3.SR3). As we will see shortly, examination of the individual tables is essential to determine the possible cause of lack of fit. In this case, since we have no good “biological explanation” for TEST3.SR3 (obviously, since this is a simulated data set!), we accept the general lack of significance of the other tables, and conclude that there is no evidence over all occasions that “survival” differs between

newly marked and previously marked individuals.

- Now let’s look at TEST3.Sm2 (i.e., TEST.3Sm for occasion 2). Recall that this test focuses on “of those individuals seen again, when were they seen again, and does when they were seen differ among previously and newly marked individuals?”. As with TEST3.SR, there is a contingency table for each of the batches, starting with the second occasion, and ending with occasion 4.
- Why not occasion 5? Well, think about what TEST3.Sm is doing - it is comparing when individuals are seen again (as opposed to are they seen again). At occasion 5, any individual if seen again must have been seen again at the last occasion (6), since there are no other occasions! So, it doesn’t make much sense to create TEST3.Sm for the penultimate occasion.
- Let’s consider TEST3.Sm2 - the second occasion.

Goodness of fit test of seen before versus not seen before against when next seen again by capture occasions.

**Test for Group 1
Control Group**

TEST 3.Sm2: Animals captured on occasion 2

O!	47	20	67
E!	47.6	19.4	
C!	0.0	0.0	
O!	17	6	23
E!	16.4	6.6	
C!	0.0	0.1	
	64	26	90
	Chi-square=0.1181 (df=1)		P=0.7311

- At occasion 2, a total of 191 individuals were released - 41 that had been seen before, and 150 newly marked individuals. Of these 191 individuals, 90 were seen again. From the TEST3.Sm2 table, 67 of this 90 were previously marked individuals, and 23 were newly marked.

You should be able to determine where these totals come from, using the full m -array.

- However, we're now faced with a different puzzle - why only two columns? If TEST3.Sm considers "when" individuals were seen again, then unless all individuals seen again were seen on only the next two occasions, then there should be more than two columns.
- Look at the full m -array. We see that of the 41 individuals previously marked and released at occasion (the {11} individuals), 23 were seen again, 17 at occasion 3, and 6 at occasion 4. So, two occasions - two columns. But consider the newly marked {01} individuals. Of the 150 newly marked individuals released at occasion 2, 67 were seen again - 47 at occasion 3, 16 at occasion 4, 2 at occasion 5 and 2 at occasion 6. So, for {01} individuals, 4 occasions, and 4 columns. Thus, if we were to construct our own TEST3.Sm2 table, it would look like:

TEST3.Sm2	when seen again?				
	seen at (2)	(3)	(4)	(5)	(6)
{11}		17	6	0	0
{01}		47	16	2	2

- So why doesn't the RELEASE table for TEST3.Sm2 look like this? It doesn't because RELEASE is "smart" enough to look at the "true" table (above) and realize that the data are too sparsely distributed for occasions 5 and 6 (0 and 0 for {11} and 2 and 2 for {01} individuals, respectively). Thus, on our own we would probably want to pool the few individuals seen for the first time at occasions 5 and 6 into the occasion 4 cell yielding the table we saw in the RELEASE output file:

TEST3.Sm2	when seen again?		
	seen at (2)	(3)	(4)
{11}		17	6
{01}		47	20

- RELEASE simply does this pooling for us, whenever "it thinks" the data are too sparsely distributed. In general, unless both survival and recapture rates are very high, RELEASE will often collapse everything down into a simply 2x2 contingency table for TEST3.Sm, meaning that for occasion (i), only occasion ($i+1$) and ($i+2$) will be in the table.
- In this case, $\chi^2 = 0.118$ (df-1), $P=0.731$. In other words, no significant difference. Overall, for this simulated data set, TEST3.Sm was not significant ($P=0.926$). Further, the overall TEST3.S result (TEST3.SR + TEST.3Sm) was also non-significant ($P=0.294$).
- Now consider TEST2, starting with TEST2.Ct. Recall that in TEST2.Ct, we are "using" individuals that are known to have survived from (i) to ($i+1$). TEST2.Ct tests if the probability of being seen at occasion ($i+1$) is a function of whether or not the individual was seen at occasion (i), conditional surviving from (i) to ($i+1$). TEST2 differs from TEST3 somewhat in that we are not considering when an individual was *marked*, but rather on when it was *recaptured*.
- The result for TEST.2Ct2 is shown on the next page.

```

Goodness of fit test of not seen versus seen just before
against next seen immediately versus next seen later

      Test for Group 1
      Control Group

TEST 2.Ct2: Test of row 1 vs. row 2

+-----+-----+
O:  9  |  4  | 13
E:  9.2|  3.8|
C:  0.0|  0.0|
+-----+-----+
O:  64 |  26 | 90
E:  63.8| 26.2|
C:  0.0 |  0.0 |
+-----+-----+
              73   30   103
Chi-square=0.0195 (df=1) P=0.8891
Fisher's Exact Test P=1.0000
    
```

```

Goodness of fit test of not seen versus seen just before
against when next seen later by capture occasion

      Test for Group 1
      Control Group

TEST 2.Cm2: Test of row 1 vs. row 2

+-----+-----+
O:  4  |  0  |  4
E:  3.5|  0.5|
C:  0.1|  0.5|
+-----+-----+
O:  22 |  4  | 26
E:  22.5| 3.5|
C:  0.0 |  0.1|
+-----+-----+
              26   4   30
Chi-square=0.7101 (df=1) P=0.3994
Fisher's Exact Test P=0.6205

** WARNING ** One or more expected values were < 2.0.
    
```

- These values can be obtained directly from the reduced m -array. Take the top row. At occasion 2, there were 13 individuals out of the total 54 recaptures for the first release batch ($r_1=54$) which were NOT caught at occasion 2. In other words, $13 = r_1 - m_{1,2}$. This value is indicated below each column in the reduced m -array, as z_i . So, at occasion 2, 13 individuals which (a) had been marked at occasion 1, and (b) were known to have survived to occasion 3, were NOT seen at occasion 2. These individuals are compared with the 90 individuals recaptured from the second release batch ($r_2=90$) - in other words, newly marked individuals at occasion 2. Of these 90 individuals, 64 were recaptured at occasion 3 (i.e., $i+1$), and 26 were captured later.
- For this occasion, the difference between the previously marked and newly marked individuals in their "immediate recapture rates" was not significant ($\chi^2=0.020$ (df=1), $P=0.889$). Overall, TEST2.Ct was not significant ($P=0.094$).
- Finally, we look at TEST2.Cm. In this test, we're simply asking the question analogous to TEST3.Sm - if seen again, when were they seen again, conditional on not being seen at occasion ($i+1$)? Here is the result for TEST2.Cm2.

- In the first row, there were 4 individuals marked at occasion 1, not seen at occasion 2, and not seen immediately at occasion 3 (i.e., they were seen at occasion 4, 5 or 6). In the second row, there were 26 individuals released at occasion 2 which were not seen immediately at occasion 3, but were seen at either occasion 4, 5 or 6. For this occasion, TEST2.Cm2 was not significant ($\chi^2=0.710$ (df=1), $P=0.399$). However, note the warning - RELEASE is telling you that one or more of the expected values was < 2.0 . Significances from tables with sparse or "0" cells should be viewed cautiously. The overall result for TEST2.Cm was not significant ($P=0.592$). The total TEST2.C tests (TEST2.Ct + TEST2.Cm) was also not significant ($P=0.190$).
- RELEASE then presents you with a convenient tabulation of all of the individual TEST3 and TEST2 results. It also gives you some indication as to whether or not there was sufficient data in a given test for you to be able to "trust" the result. In this case, there seems to be insufficient data for TEST2.Cm - we then have to decide on whether or not we "believed" the results.

Summary of TEST 3 <Goodness of fit> Results					
Group	Component	Chi-square	df	P-level	Sufficient Data
1	3.SR2	1.6885	1	0.1938	Yes
1	3.SR3	6.9840	1	0.0082	Yes
1	3.SR4	0.0322	1	0.8576	Yes
1	3.SR5	0.0122	1	0.9120	Yes
Group 1	3.SR	8.7170	4	0.0686	
1	3.Sm2	0.1181	1	0.7311	Yes
1	3.Sm3	0.1851	1	0.6670	Yes
1	3.Sm4	0.5846	2	0.7465	Yes
Group 1	3.Sm	0.8878	4	0.9263	
Group 1	TEST 3	9.6048	8	0.2939	

Summary of TEST 2 <Goodness of fit> Results					
Group	Component	Chi-square	df	P-level	Sufficient Data
1	2.Ct2	0.0195	1	0.8891	Yes
1	2.Ct3	1.7868	1	0.1813	Yes
1	2.Ct4	4.5795	1	0.0324	Yes
Group 1	2.Ct	6.3857	3	0.0943	
1	2.Cm2	0.7101	1	0.3994	No
1	2.Cm3	0.3384	1	0.5608	No
Group 1	2.Cm	1.0484	2	0.5920	
Group 1	TEST 2.C	7.4342	5	0.1903	

Goodness of Fit Results <TEST 2 + TEST 3> by Group

Group	Chi-square	df	P-level
1	17.0389	13	0.1975

- Accepting the TEST2.Cm result as valid, we have no significant TEST2 or TEST3 result. Thus, the overall GOF result (TEST2 + TEST3 = 17.04) is also not significant (P=0.198). This is perhaps not surprising, since we set up the simulation so that the data WOULD follow the CJS assumptions! Our purpose here was simply to introduce TEST2 and TEST3.
- One thing you might be asking yourself at this point is “since RELEASE gives me these nice summary tables, why do I need so much DETAIL?”.
- The answer - if your data DO fit the CJS model, then you clearly don't. But if they don't fit the model (i.e., if any of the 4 tests is rejected), then the only chance you have of trying to figure out what is going on is to look at the individual contingency tables. We got some sense of this when we looked at TEST3.SR in our simulated data set - one of the batches had results quite different from the other batches, leading to a near-significant TEST3.SR result overall. Further, even if the 4

tests are accepted (no significant differences) you should remember that these tests are for simple heterogeneity - they do not test specifically for systematic differences. Again, the only clue you might have is by looking at the individual tables.

What to do when CJS (ϕ, p_i) is rejected?

- There are at least 2 classes of reasons why CJS might be rejected by RELEASE. The first is the “biologically interesting” one - specifically, that the CJS model is simply inappropriate for the data. Beginning with Chapter 7, you've seen how we might have to radically alter the basic ultrastructure of the CJS model to accommodate “biological reality”. For example, if you are marking both juveniles and adults, then there is perhaps a reasonable expectation that their relative survival rates may differ, and thus, one of the assumptions of CJS (specifically assumption 2) - that every **marked** animal in the population immediately after time (i) has the same probability of surviving to time ($i+1$) has been violated.
- The second, perhaps less interesting reason is the possibility that the data are over or under-dispersed for the CJS model - extra-binomial variation.
- In this section, we will briefly discuss both cases.

a. *inappropriate model*

- To confirm that CJS has been rejected because the model is “biologically unrealistic” for your data, you will need to carefully examine the detailed TEST 2 and TEST 3 contingency tables in your RELEASE output file. What are you looking for? Well, in general, the thing you're looking for is a “systematic” rejection (or bias) in the individual tables. You need to see if the failure of TEST2 or TEST3 is “driven” by a few “strange” batches, or is due to a “systematic” bias.
- What do we mean by “systematic” bias? Well, by “systematic”, we refer to a bias which occurs consistently at each occasion - a bias in the sense that a particular cell (or cells) in one of the test tables is

consistently over or underpopulated.

- An example will help make this clear. Suppose you run RELEASE, and find that TEST 3 is rejected, but not TEST 2. You say to yourself, “OK, recapture seems to be OK, but something is wrong with the survival assumption, under the CJS model”. You proceed to look carefully at each of the TEST3 tables for each batch. You note that TEST3.SR is rejected, but that TEST3.Sm is accepted. Now, what does this mean? Recall that TEST3.SR simply asks, of those individuals seen either on or before occasion (*i*), what proportion were ever seen again?
- If TEST3.SR is rejected, then this suggests that there is a difference in “survival” among individuals, depending on whether or not they were seen for the first time either on or before occasion (*i*).
- However, TEST3.Sm only looks at individuals who WERE seen again. Among these individuals, *when* they were seen again does not depend on whether or not they were seen for the first time at occasion (*i*).
- Suppose you look at each individual TEST3.SR table, and find the following - a “+” indicates more individuals than expected (based on the null hypothesis of no differences between groups), and a “-” indicates fewer individuals than expected (under the same hypothesis). Since RELEASE provides you with the expected frequencies under the null hypothesis, you’ll be able to make this diagnosis very quickly.

<i>TEST3.SR</i>		
seen at (<i>i</i>)	seen again	not seen again
seen before	(+)	(-)
not seen before	(-)	(+)

- Let’s say we have 10 occasions, and we find that this pattern seems to be present in the majority of them (you might use some statistical test, for example a sign test, to determine if the frequency of tables exhibiting a particular pattern occurs more often than expected by random chance). Say, 8/10 contingency tables show this pattern. What

does this suggest? Well, among individuals seen for the first time at occasion (*i*), significantly more are never seen again than expected, relative to individuals who had been seen before occasion (*i*). In other words, newly marked individuals showed a consistently lower probability of ever being seen again than previously marked individuals.

- What could lead to this pattern? One possibility we suggested at the beginning of this section was age effects. Lower survival of newly marked juveniles (relative to adult survival) would lead to this pattern in TEST3.SR.
- Is it the “only” plausible explanation? Unfortunately, no. Life would be simpler if there was only ever one explanation for anything, but, this is generally not the case. This example is no exception. Rejection of TEST3.ST could also reflect (a) a marking effect (where the act of marking causes an increase in immediate mortality), (b) presence of transients (migratory individuals leaving the sampling area shortly after marking), or (3) heterogeneity in capture rates (some individuals have low capture rates, some high).
- The point here is that there may be more than one possible answer - it is at this point you’ll need to use your “biological insight” to help differentiate among the possible explanations. The other, perhaps more important point is that the presence of a consistent difference in one of the 4 major tests (TEST2.Ct, TEST2.Cm, TEST3.SR, and TEST3.Sm) each suggest the possibility of one or more effects which violate the basic CJS assumptions. You will have to rely on your insight to help you identify possible “biological” explanations for violation of any of these 4 tests - each of them might refer to something completely different. For example, if there is systematic bias in TEST2.Ct, but acceptance of TEST3.SR, TEST3.SM, and TEST2.Cm, the sum of these three tests is a GOF test for a model with immediate trap-dependence in recapture. Can you think of why?
- What can you do if you do reject CJS? Well, the solution depends on what, exactly, “has gone wrong”. In general, if the individual TEST2 or TEST3 results seem to show systematic deviations among occasions, the most likely solution will be to reject the CJS model as the “correct” starting model for your analysis - it clearly doesn’t fit, because the inherent assumptions aren’t met by the data. In this case,

where TEST3.SR is rejected, but the other tests are accepted, then the solution is to add age-structure to the model (see Chapter 7).

- However, simply recognizing that a “different “starting model (say, a 2-age class model) may be more appropriate is only the first step. You still need to confirm that the data fit your “new” model. You must go through analogous GOF tests for the “new” starting model, just as we have done for the CJS model.
- Now, the question is, if RELEASE only tests the CJS model, how can we do GOF testing for non-CJS models? The answer is, you can use RELEASE for other models, under some conditions.
- One possibility is that you can use summations of some of the 4 tests to test specifically for other models. For example, if TEST3.SR is biased, but the other 3 tests are accepted, then the sum of (TEST3.Sm + TEST2.Ct + TEST2.Cm) represents a GOF test for model ϕ_{a_2, t, p_t} (note: the df are also summed)! Can you figure out why? With a bit of work, you should see the logic. If not, don't feel too bad - this (and other) RELEASE “tricks” haven't even been published yet!. What is worth noting is that this “trick” simply makes use of the additive properties of χ^2 statistics - if TEST3.SR is violated, it is OK to sum up the other test statistics to create a GOF for the model for which TEST.3SR means little, like ϕ_{a_2, t, p_t} !
- Another option is also available if you have plenty of data - enough so that you can run RELEASE for each batch independently.
- Why is this useful? Well, if you remember back to Chapter 7 where we introduced age and cohort models, you may recall that, within a cohort, age and time are collinear (since cohort = time - age). As such, testing for “age” and “time” effects within cohort are synonymous. Then, the sum of the individual cohort GOF statistics makes a global GOF statistic for model $\phi_{a^*t, p_{a^*t}}$

b. Extra-binomial variation

- What do you do, though, if the deviations in the contingency tables are not “consistent” among batches - what if (say) 5/10 tables are biased in one direction, and 5/10 are biased in the other direction?
- In such cases, where there seems to be no clear “explanation”

(biological or otherwise) for the violation of TEST2 or TEST3, you then have only a few options. The most common “solution” (although it is only a partial solution) is to “adjust” your statistics to account for the “extra noise”, or (for the statistically inclined) the extra-binomial variation.

- Remember that the conceptual basis of all models is “data = structure + residual variation”. In general, the structure of the residual variation is unknown, but for multinomial distributions, it IS known. If the model structure is “correct”, then the variance of the residual “noise” is 1 (where variance is defined as the expected value of the GOF χ^2 divided by its degrees of freedom).
- However, even if the residual variation > 1, the structural part of the model can be “correct”. In simplest terms, if there is “excess variation” it will show up in the model GOF χ^2 (since this value is essentially a “residual SS”).
- Thus, what we need to do is “correct” everything for the magnitude of this extra variation. To do this, we derive what is known as a variance inflation factor, c . The larger the value of c , the greater the amount of “extra” variation.
- How do we do this? Actually, RELEASE makes it easy. Simply take the sum of the TEST2 and TEST3 χ^2 values, and divide by the overall CJS model degrees of freedom (i.e., the number of estimable parameters in the model).

$$c = (\text{TEST2} + \text{TEST3})/\text{df}$$

- So what do we do with our “correction factor”, c ? We use it to adjust the test statistics (and the standard errors of the estimates) used for comparing the fit of different models to the data - the likelihood ratio test (LRT), and the Akaike Information Criterion (AIC).
- Here (without proof) are the transformations for the LRT and AIC, respectively:
- LRT - the LRT, which is normally a χ^2 test of fit between models, is transformed into an F-test ($\text{df}=\text{df}_{\text{LRT}}, \text{df}_{\text{CJS}}$) as:

$$F = \frac{C_{LRT}^2 / df_{LRT}}{c}$$

- For the AIC, which is estimated for a given model as $AIC = (\text{model deviance}) + 2(\text{\#parameters})$, we simply divide the model deviance by c , such that:

$$AIC = \frac{DEV}{c} + 2 * np$$

- At what point is c too large to be useful? If the model fits perfectly, then $c=1$. What about if $c=2$, or $c=10$? Is there a point of diminishing utility? As a working “rule of thumb”, provided $c \leq 3$, you should feel relatively safe (see Lebreton *et al.* 1992 - pp. 84-85).

That's the end of our **very** quick stroll through the problem of GOF testing in mark-recapture analysis. In this appendix, we have focussed on using program RELEASE to handle this task. We're the first to admit that, at some levels, this whole process is a bit more complex than we would like. Ideally, we'd like an application that took ANY arbitrary model we fed into it, and did the GOF testing in a robust and intelligent fashion. To some degree, it could be argued that program SURVIV does something like this already. However, SURVIV does not provide nearly the amount of useful detail that RELEASE does - the statistics provided by SURVIV only point out that there "is" a problem - they offer little guidance as to "why", or "where" the problem might occur. Such an application would be much appreciated by people working with CMR analyses, including us!