First steps in using Program MARK...

In this chapter we will introduce the basic mechanics of running program **MARK**, using a small data set consisting of 7 years of capture-recapture data on a small passerine bird, the European dipper (*Cinclus cinclus*). This data set is the same as that used in 'Examples' in Lebreton *et al*. (1992), and consists of marking and recapture data from 294 breeding adults each year during the breeding period, from early March to 1 June.[*] All birds in the sample were at least 1 year old when initially banded. We'll forgo discussion of GOF (Goodness of Fit) testing for the moment, although we emphasize that, in fact, this is the prerequisite step before you analyze your data. GOF testing is covered later.
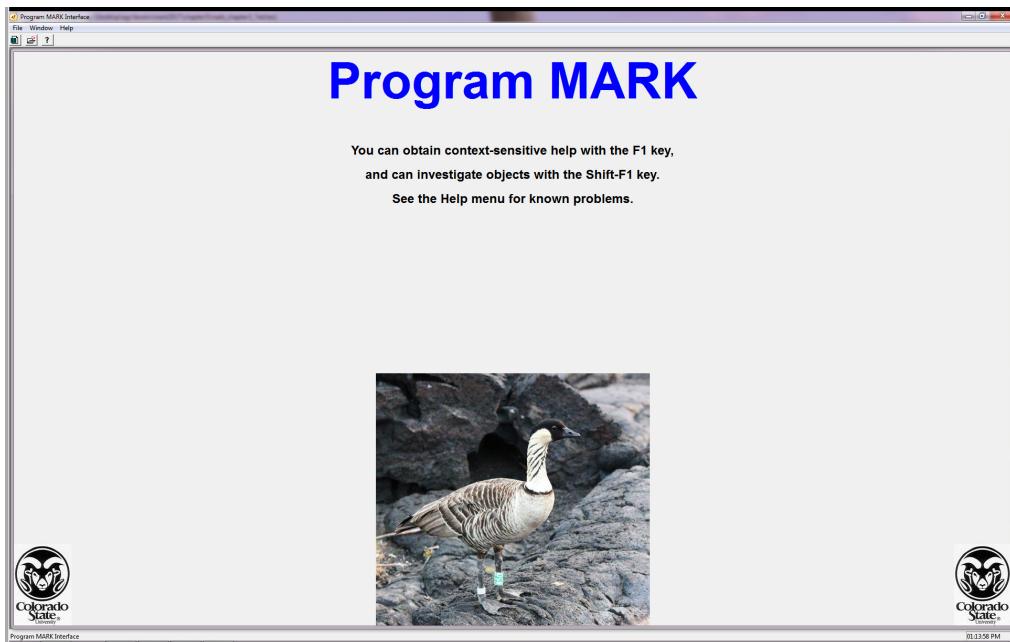
Our main intent here is to show you the basics of running **MARK**, not to provide great detail on 'why you are doing what you are doing'. Our experience has shown that perhaps the greatest initial hurdle to using a new piece of software, especially one as sophisticated as **MARK**, is the 'newness' of the interface, and the sheer number of options available to the user. In addition, we are starting with the dipper data set, since it has been extensively analyzed in several places, and will be very familiar to many experienced users migrating to **MARK** from another application.

We also believe that starting with a 'typical' mark-recapture problem is a good place to begin – if you understand how to do a mark-recapture analysis, you're much of the way to understanding the principles behind many of the other analytical models incorporated into **MARK**. The male subset of the dipper data set is not one of those which are 'bundled' with **MARK** (although the full dipper data set is). We'll assume here that you've managed to extract the dipper data set **ed_males.inp** from the **markdata.zip** file that accompanies this book (if you don't have **markdata.zip**, you can download it from the same website you downloaded this book from).

3.1. Starting MARK

Since **MARK** is a true Windows application, starting it is as simple as double-clicking the **MARK** icon (which we assume resides in some specified folder on your desktop). Locate the icon, and double-click on it. **MARK** is a fairly large program (although this is now a relative statement with the advent of 100+ MB word processors!), and may take a few moments to start up. If all goes well, you should soon be presented with the opening 'splash screen' (shown at the top of the next page – the particular 'warm and fuzzy organism' you see will depend on which version of **MARK** you are using), indicating that **MARK** is 'up and running'. If nothing happens (or you get some typically obscure Windows error message), this is a good indication that something is not working right.

---
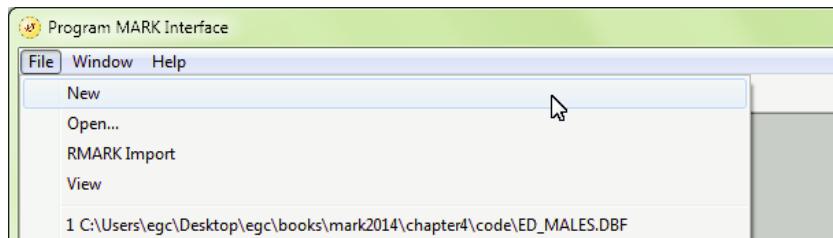
[*] And is, without question, the most re-analyzed set of mark-recapture data in the world.

Last updated: 12.19.2025
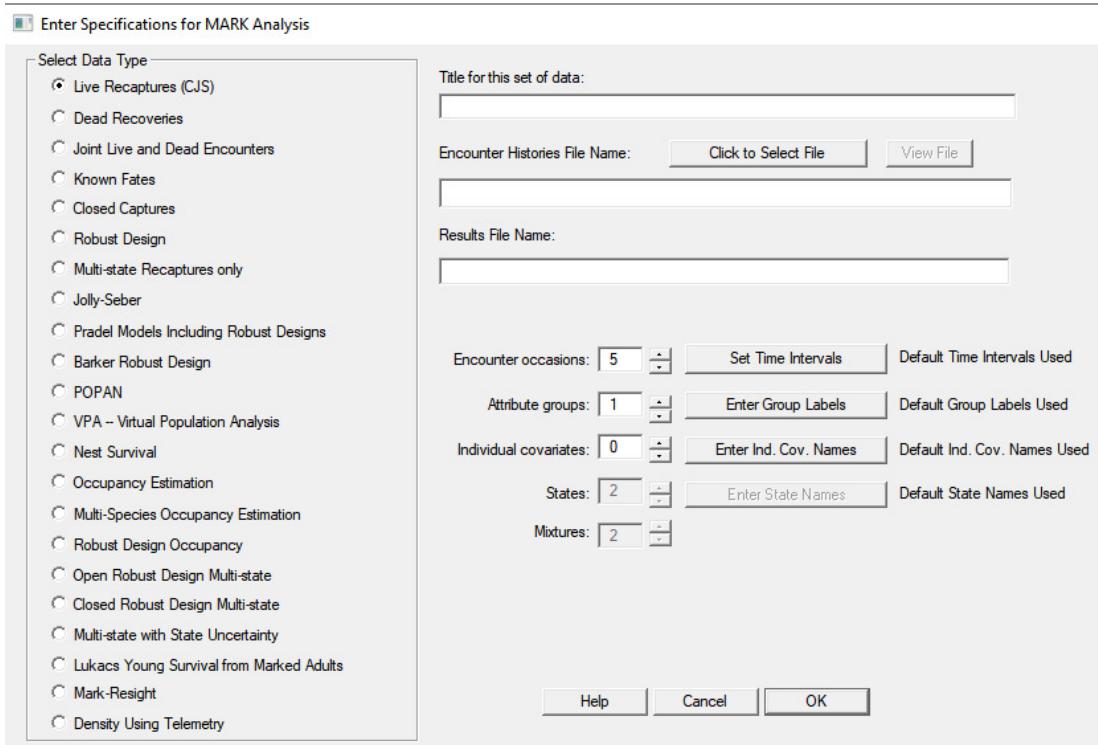
Unfortunately, figuring out the problem depends to a large degree on things like: how many other applications do you have open? Are you sure you've installed the most recent version of **MARK**? How much memory is in your machine? Are you running Windows 11? Windows 10? And so on. **MARK** is very robust on most machines, so problems getting it started should be very infrequent. If it doesn't start correctly, then try again after first closing all other running applications (in general, **MARK** runs 'best' when it is the only program running). If that doesn't work, then trying re-installing from scratch – if you've already downloaded the `setup.exe` file, then this should only take a few moments (again, when in doubt, try reinstalling – this is often a good way to 'fix' minor problems that might arise).

3.2. starting a new project

The very first thing you need to do is to tell **MARK** you're going to start a new project (we're assuming that at this stage, you don't have any existing **MARK** projects going). Doing this is very easy – all you need to do is pull down the '`File`' menu in the upper left-hand corner of the main **MARK** window, and select '`New`' from the drop-down menu:

Once you have selected the '`New`' option from the drop-down '`File`' menu, the graphical 'splash-screen' that you saw when you started **MARK** will be erased, and you will be presented with a new sub-window – the specification window for program **MARK** (shown at the top of the next page):

Chapter 3. First steps in using Program MARK...
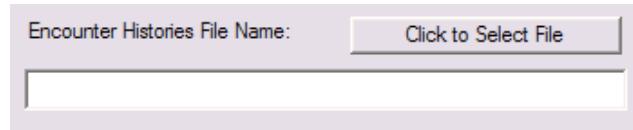
Clearly, the specification window contains a fair bit of information, but the 'basics' can be broken down into 4 main sections.

First, on the left of the window (highlighted by a bezelled line) is a simple radio-button list for the various data types **MARK** can handle. In fact, this is the point at which you tell **MARK** what kind of analysis you want to do. In its simplicity, this list belies the sheer scope of analytical coverage provided by **MARK**. Whereas most previous applications specialized on one (or a couple) of types of analysis (for example, live recapture-only, or dead recovery-only), **MARK** can handle most of the common analytical designs in use today. While **MARK** is clearly not a replacement (in some ways) for a more general purpose approach like **SURVIV** (or, more recently, **R**, **MATLAB** or **JAGS**), it **is** in many respects a replacement for much (if not all) of the 'canned' software previously in general use. The '`live recaptures (CJS)`' data type is selected by default – and, is the data type we want for this analysis. At the top of the right-hand half of the window is a fill-in element for the '`Title`' of the project. This is available as a convenience to the user, and does not have to be filled out. For this example, we use a title reflecting the fact we're analyzing the male data from the European dipper study:
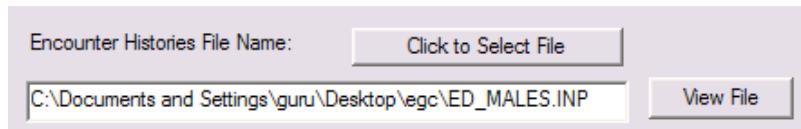
Immediately below the title field is a second fill-in element where the user specifies the file containing the encounter histories they want to analyze. If the file name and path are known, they can be entered directly into this box. More typically, you will want to browse for the file (i.e, select it manually), by clicking on the '`Click to Select File`' button immediately below the box (you can also double-click
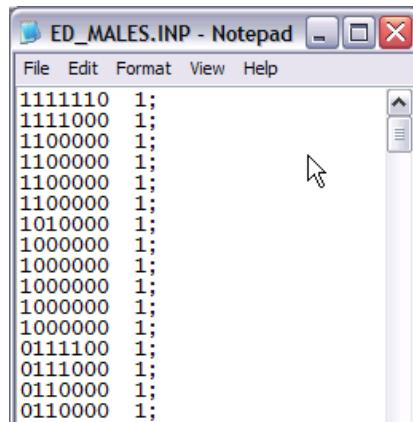
the fill-in box itself). If you click on this button, you will be presented with the standard Windows interface to finding and selecting a particular file (see below).

One thing to note is that until you have selected a file to analyze, the '**View File**' button is not available (i.e., is not active). Once you have selected the file, you will be able to view its contents. This is useful if you forget certain things about your data (for example, the number of occasions). In our example, we select the file **ed_males.inp** (obviously, the path shown here reflects the machine we're running **MARK** on, and will **not** be the same as what might appear on your machine).
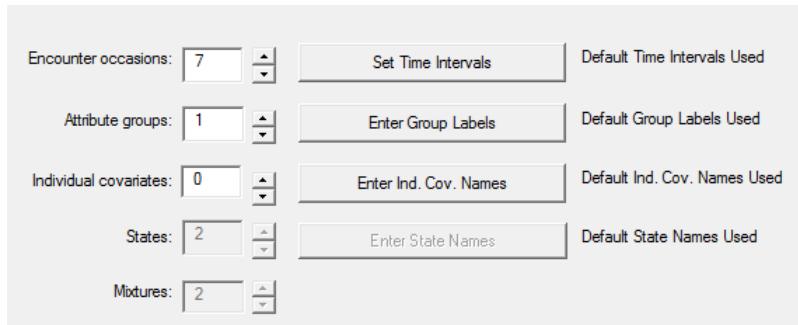
Again, note that once the file has been specified, the '**View File**' button become active. If we click on **View File**, **MARK** starts up the default Windows browser (usually the Windows **Notepad** – if your file is too large for **Notepad** to handle, then Windows will prompt you to use an alternative application to view the file). We see from the **Notepad** window (pictured below) that we have 7 occasions, and only 1 group (recall from Chapter 2 the basics of formatting data for processing by program **MARK**).

Finally, the bottom-half of the right hand side. Here, you specify the number of sampling occasions in the file. Because the `.INP` files containing the capture histories do not explicitly code for the number of occasions, you will have to tell **MARK** how many occasions there are. It defaults to 5 – do **not** be fooled into thinking this is the number of occasions in your file. **MARK** has no way of knowing what this value is – you have to enter it explicitly. You can also tell **MARK** how many '**attribute groups**' are in the data (for example, if your file contained data for both males and females, you would enter 2). Finally, the number of individual covariates. (*Note*: the boxes for the '**number of strata**' and **Mixtures**' only becomes available if you select the appropriate data types which require specifying these options.)

To the right of each input box, there is a button which allows you to control some aspects of each box. For example, to the right of the number of occasions box is a button which lets you set the intervals between each sampling occasion. The default is '`1 time period between each occasion`'. We'll talk more about this particular option later. The other two buttons for '`group labels`' and '`individual covariate`' names are (we suspect) fairly self-explanatory. For this analysis, since we have only one attribute group (males), there is no real need to specify a particular label.
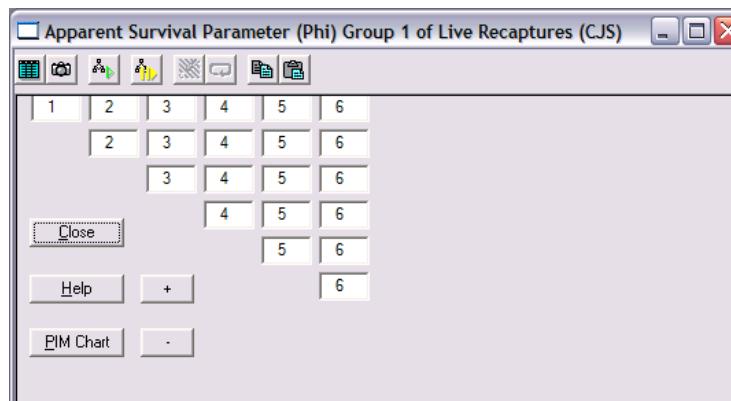
In our dipper example, we have 7 occasions, and only 1 group (as shown below):

Now, once you've got everything set in the specification window, you're ready to proceed (if not, simply correct the individual entries as needed, or cancel to start over). To continue, simply click the '`OK`' button.
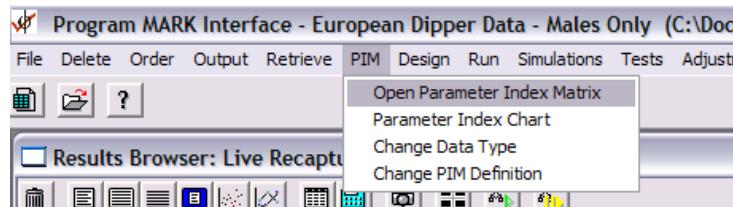
The first thing that will happen when you press the '`OK`' button (assuming that you've specified a file that exists) is that **MARK** will close the specification window, and present you with a small little pop-up window telling you that it has created a DBF (data-base format) file to hold the results of your analyses. The name of the file in this example, is **ed_males.dbf**. **MARK** uses the prefix of the `.INP` data file (in this case, **ed_males**) as the prefix of the DBF file (resulting in **ed_males.dbf**). **MARK** pauses until you press the '`OK`' button, telling it to proceed. If **ed_males.dbf** already existed (i.e., if you'd already done some analysis on these data), **MARK** would inform you that it will have to overwrite the existing file, and will then ask you if this is OK.

Since we haven't run any analysis on these data before, we simply click the '`OK`' button. Doing so causes the pop-up window to close, and a window containing a 'triangular' matrix representation of the survival model structure is presented:
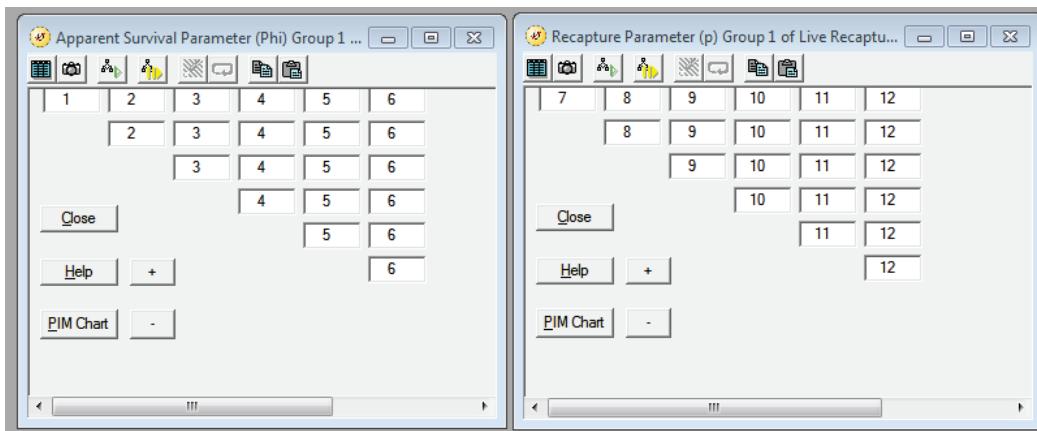

However, you'll note from the title of the window that the matrix represents only the survival

parameters. For a mark-recapture analysis, we're also interested in estimating the recapture probability. What you don't see here is that, by default, **MARK** initially presents you with only the survival parameters, assuming (presuming?) that this is what you're most interested in working with. However, clearly we may (and probably should) also be interested in modelling the other parameters which define the system (in this case, the recapture parameter). For now, let's get **MARK** to show us both parameter matrices. We get **MARK** to show us any (or all) of the parameter matrices by accessing the '**Parameter Information (or Index) Matrix**' menu (PIM), and selecting the '**Open Parameter Index Matrix**' option:

Once you select this option, you will be presented with a new window containing a list of parameters which you can 'open' – in other words, a list of parameters you can ask **MARK** to show you in the 'triangular matrix' format. In our example, since the survival parameter chart is already open, the only one which will show up on the list is the recapture parameter. You can either select it individually, or use the '**Select All**' button. Once you have selected the recapture parameter matrix, you simply click '**OK**', to cause **MARK** to place the recapture PIM in its own window. Initially, they may (or likely will) be presented in an overlapped way (technically known in Windows-jargon as 'cascaded'). To see the 2 PIM windows separately, you can access the '**Window**' menu, and select '**Tile**'. This will cause the PIM windows to be arranged in a 'tiled' (i.e., non-overlapping) fashion.
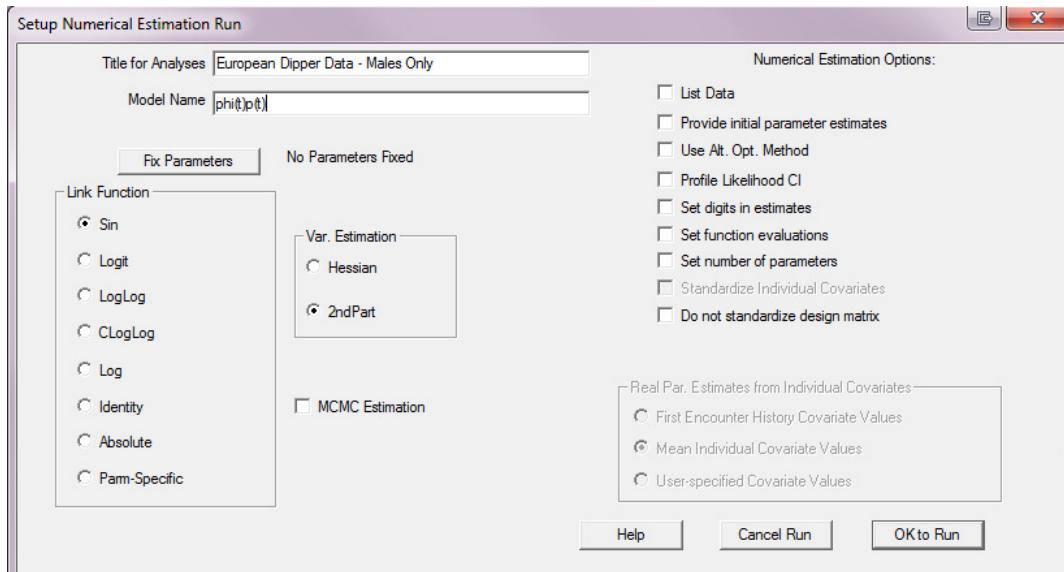
If you've had some background in mark-recapture analysis, you might recognize that these PIM's reflect the fully time-dependent Cormack-Jolly-Seber (CJS) model. If you're new to mark-recapture analysis, not to worry – all of this gets covered in great detail in subsequent chapters. For the moment, all you need to know is that in this analysis, the survival parameters are numbered 1 to 6 (there are seven occasions, so six intervals over which survival is estimated), and 6 recapture parameters (numbered 7 through 12). In fact, there are only 10 separable parameters and one 'product' parameter estimated in this analysis (11 total parameters), but we'll discuss the details concerning this later.

Chapter 3. First steps in using Program MARK. . .

3.3. Running the analysis

Let's proceed to run the analysis. Not surprisingly, to run an analysis in **MARK**, you need to access the '**Run**' menu. In this example (and at this stage), when you access the '**Run**' menu, **MARK** will present you with several options – for now, we want to select the '**Current Model**' option.

The first thing that **MARK** does is present you with a new window – the '**Setup Numerical Estimation Run**' window (since this is a rather awkward window name, we'll refer to it simply as the '**Run**' window):

Much like the specification window we saw earlier, the '**Run**' window (above) contains a lot of options, so let's take some time at this point to get you familiar with the components of this window. As with the specification window, the run window can be broken down into 4 major elements. First, in the top left, the analysis title (which **MARK** carries over from what you entered earlier in the specification window), and the model name (which is initially blank). **MARK** must have a model title to refer to (the reason why will be obvious later), so we need to enter something in the model name input box. Following the naming convention advocated in Lebreton *et al.* (1992) (which follows closely from standard linear models notation), we'll use 'phi(t)p(t)' as the name for this model, reflecting the fact that the model we're fitting has full time-dependence for both survival (φ – since **MARK** doesn't allow you to enter non-ASCII text, we're restricted to writing out the Greek symbol as 'phi') and recapture (p). The (t)-notation simply indicates time-dependence. So, $\{\varphi_t\, p_t\} \rightarrow$ 'phi(t)p(t)'.
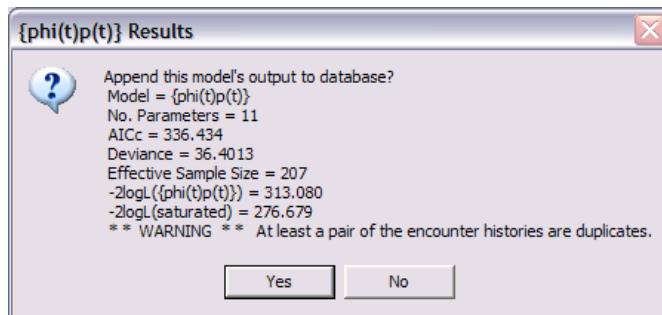
Next, the lower-left hand side of the run window. This is where we specify the link function (**MARK** lets you choose among several different functions – the sin link is the default, while the logit link is perhaps the most familiar). We'll talk about link functions in more detail later. For the moment, we'll use the default sin link.

To the right of this list of link functions is a list of two '**Var. Estimation**' options – these are options which controls how the variance-covariance matrix is estimated. This is important because this estimation provides both the information needed to compute the standard errors of the estimates, but is also used to calculate the number of estimable parameters in the model (in this example, there should be 11 estimable parameters). The '**2nd Part**' option is the default and preferred option, so we'll use it here. Finally, just above the link and variance estimation lists is a button which allows you to '**fix parameters**'. As we'll see in subsequent chapters, there will be occasions when we need to specify a value

(normally 0 or 1) for some parameters – in this example, there is no need to do so. Fixing parameters that are logically constrained to be a particular value is useful, since it both reduces the number of estimable parameters, and helps **MARK** estimate the remaining parameters more accurately and efficiently.

On the right-hand side of the run window are a series of program options related to the numerical optimization. Most of the options are fairly self-explanatory, so we won't go into any of these in detail here. For now, we'll leave all of the check boxes on the right-hand side blank.

You're now ready to run the program. To do this, simply click on the '**OK to run**' button in the lower right-hand corner of the run window. If you are using the default preferences for **MARK** (which you probably will be at this point), **MARK** will spawn a box asking you if you want to use the 'identity matrix' (since none was specified). The use of the identity matrix will be covered in much more detail in Chapter 6 – for now, simply accept the 'identity matrix', by clicking '**Yes**'. Immediately after doing so, **MARK** spawns a numerical estimation window (black background). This window, which is a shell to the numerical estimation routines in **MARK**, allows you to watch the progress of the estimation. Several things will scroll by pertaining to the estimation. Much of the information being printed to the estimation window can be also printed out to the results .DBF file, by specifying the appropriate options in the run window. You may also notice that **MARK** has created another window, called the '**Results Browser**' – more on this window in a moment. Once the estimation is complete (which should only take a second or two for this dataset), **MARK** will shut down this command window, and then present you with the 'pop-up' window shown below:

{phi(t)p(t)} Results ✕

?   Append this model's output to database?
Model = {phi(t)p(t)}
No. Parameters = 11
AICc = 336.434
Deviance = 36.4013
Effective Sample Size = 207
-2logL({phi(t)p(t)}) = 313.080
-2logL(saturated) = 276.679
** WARNING ** At least a pair of the encounter histories are duplicates.

[Yes]   [No]

This window provides summary of some of the key results of your estimation (the number of parameters estimated, the model deviance and so forth). The purpose of this summary is to give you the option of whether or not to append the full results of the estimation to the result database file (DBF). In theory, you could decide at this point that there was something 'wrong', and not append the results. You might wonder about the **WARNING** statement at the bottom – if you read it carefully, you'll see that **MARK** is 'warning' you that at least one pair of encounter histories in the .INP file are duplicates. In fact, for .INP files where each line represents the encounter history for a different individual (as in the present example), this is entirely expected. Meaning, there is nothing to worry about in this case. So we click '**Yes**', and accept the results. As soon as you click '**Yes**', two things happen in rapid succession. First, the results summary window closes. And then, an item is added to the results browser window (below):

Results Browser: Live Recaptures (CJS)

Model	AICc	Delta AICc	AICc Weight	Model Likelihood	No. Par.	Deviance
{phi(t)p(t)}	336.4343	0.0000	1.00000	1.0000	11	36.4013

Chapter 3. First steps in using Program MARK...

3.4. Examining the results

Now that we have something to 'look at' in the browser window, we can focus a bit more on the structure of the window, and what options are available. First, in the main body of the results browser, you have several columns of information. From left to right: the model title (or simply, '**Model**'), the corrected or adjusted Akaike Information Criterion (AIC$_c$), the ΔAIC$_c$ value (the difference in the value of the AIC$_c$ from the model currently in the browser having the lowest AIC$_c$ – since we have only one model in the browser at the moment, ΔAIC$_c$ = 0), the AIC$_c$ weight, the model likelihood, the number of estimated parameters, and the model deviance. The column arrangement can be modified to suit your preferences by simply dragging the columns to the left or right. The AIC$_c$, the number of parameters and the model deviance (in particular) form the basis for comparison with other models. Since we only have one model at the moment, we'll defer discussion of these issues until later. Along the top of the browser window are a number of buttons, which you can use to access a variety of functions. More on these later.
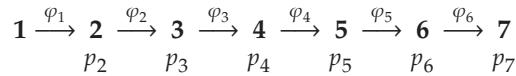
For now, we want to view the results of our estimation. We can view just the reconstituted parameter estimates (on the 'real probability scale') by clicking on the fourth button (from the left) on the menu, or by selecting '**Output | Specific Model Output | Parameter Estimates | Real**'. This spawns a Windows Notepad window, containing the estimates (shown below).

The **MARK** output consists of 5 columns: the first column is the parameter index number ($1 \rightarrow 12$), followed by the parameter estimate, the standard error of the parameter, and the lower and upper 95% confidence limits for the estimates, respectively. The parameter indexing relates to the indexing used in the PIMs we saw earlier.

3.4.1. MARK, PIMs, and parameter indexing

Let's stop a moment for a quick introduction to the indexing scheme that **MARK** uses. Consider the following figure:

$$1 \xrightarrow{\varphi_1} 2 \xrightarrow{\varphi_2} 3 \xrightarrow{\varphi_3} 4 \xrightarrow{\varphi_4} 5 \xrightarrow{\varphi_5} 6 \xrightarrow{\varphi_6} 7$$
$$p_2 \qquad p_3 \qquad p_4 \qquad p_5 \qquad p_6 \qquad p_7$$
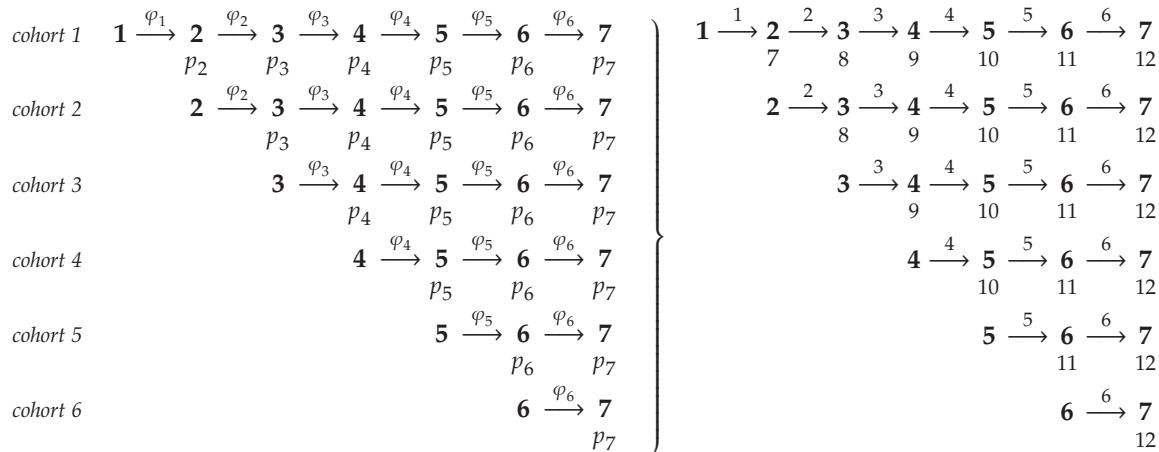
In our analysis of the male dipper data, recall that we have 7 occasions – the initial marking occasion, and 6 subsequent recapture (or resighting) occasions. The φ_i values represent the survival probabilities between successive occasions (i.e., φ_i is the probability of surviving from occasion (i) to occasion ($i+1$)), while the p_i values represent the recapture probabilities at the (i)th occasion. For details, see Lebreton *et al.* (1992).
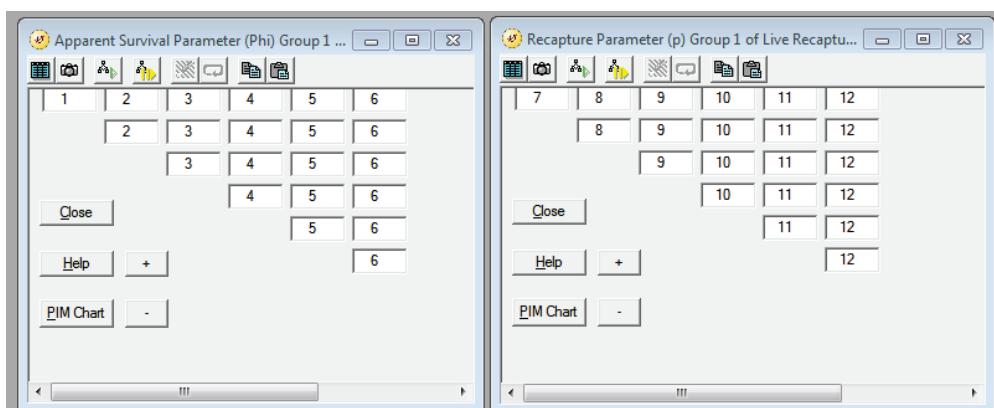
What **MARK** does is to substitute a numerical indexing scheme for the individual φ_i and p_i values, respectively. For example, consider the indexing for the survival parameters φ_i – there are 6 values, φ_1 through φ_6. How does this connect to the 'survival PIM'?

$$1 \xrightarrow{\varphi_1} 2 \xrightarrow{\varphi_2} 3 \xrightarrow{\varphi_3} 4 \xrightarrow{\varphi_4} 5 \xrightarrow{\varphi_5} 6 \xrightarrow{\varphi_6} 7$$
$$\quad p_2 \qquad p_3 \qquad p_4 \qquad p_5 \qquad p_6 \qquad p_7$$

$$\Downarrow$$

$$1 \xrightarrow{1} 2 \xrightarrow{2} 3 \xrightarrow{3} 4 \xrightarrow{4} 5 \xrightarrow{5} 6 \xrightarrow{6} 7$$
$$\quad 7 \qquad 8 \qquad 9 \qquad 10 \qquad 11 \qquad 12$$

What about individuals captured for the first time and marked at the second occasion? Technically, we would refer to such individuals as being in the second release *cohort* (a cohort is simply a group of individuals that bear something in common – in this case, individuals captured, marked and released on the second occasion would comprise the second cohort). This is similar for individuals captured, marked and released on the third occasion, and so forth. All we need to do to account for these different release cohorts is to start the 'indexing' at the appropriate occasion for each cohort – this is shown schematically, below:

cohort 1 $1 \xrightarrow{\varphi_1} 2 \xrightarrow{\varphi_2} 3 \xrightarrow{\varphi_3} 4 \xrightarrow{\varphi_4} 5 \xrightarrow{\varphi_5} 6 \xrightarrow{\varphi_6} 7$ $1 \xrightarrow{1} 2 \xrightarrow{2} 3 \xrightarrow{3} 4 \xrightarrow{4} 5 \xrightarrow{5} 6 \xrightarrow{6} 7$
$\quad\quad\quad\quad\ p_2 \ \ p_3 \ \ p_4 \ \ p_5 \ \ p_6 \ \ p_7$ $\quad\quad\ \ 7 \ \ \ 8 \ \ \ 9 \ \ \ 10 \ \ \ 11 \ \ \ 12$

cohort 2 $2 \xrightarrow{\varphi_2} 3 \xrightarrow{\varphi_3} 4 \xrightarrow{\varphi_4} 5 \xrightarrow{\varphi_5} 6 \xrightarrow{\varphi_6} 7$ $2 \xrightarrow{2} 3 \xrightarrow{3} 4 \xrightarrow{4} 5 \xrightarrow{5} 6 \xrightarrow{6} 7$
$\quad\quad\quad\quad p_3 \ \ p_4 \ \ p_5 \ \ p_6 \ \ p_7$ $\quad\quad\quad 8 \ \ \ 9 \ \ \ 10 \ \ \ 11 \ \ \ 12$

cohort 3 $3 \xrightarrow{\varphi_3} 4 \xrightarrow{\varphi_4} 5 \xrightarrow{\varphi_5} 6 \xrightarrow{\varphi_6} 7$ $3 \xrightarrow{3} 4 \xrightarrow{4} 5 \xrightarrow{5} 6 \xrightarrow{6} 7$
$\quad\quad\quad\quad p_4 \ \ p_5 \ \ p_6 \ \ p_7$ $\quad\quad\quad\quad 9 \ \ \ 10 \ \ \ 11 \ \ \ 12$

cohort 4 $4 \xrightarrow{\varphi_4} 5 \xrightarrow{\varphi_5} 6 \xrightarrow{\varphi_6} 7$ $4 \xrightarrow{4} 5 \xrightarrow{5} 6 \xrightarrow{6} 7$
$\quad\quad\quad\quad p_5 \ \ p_6 \ \ p_7$ $\quad\quad\quad\quad\quad 10 \ \ \ 11 \ \ \ 12$

cohort 5 $5 \xrightarrow{\varphi_5} 6 \xrightarrow{\varphi_6} 7$ $5 \xrightarrow{5} 6 \xrightarrow{6} 7$
$\quad\quad\quad\quad p_6 \ \ p_7$ $\quad\quad\quad\quad\quad\quad 11 \ \ \ 12$

cohort 6 $6 \xrightarrow{\varphi_6} 7$ $6 \xrightarrow{6} 7$
$\quad\quad\quad\quad p_7$ $\quad\quad\quad\quad\quad\quad\quad 12$

Now, have another look at the PIMs for survival and recapture:

The numbers 1 to 6 in the PIM correspond to the survival values φ_1 to φ_6, respectively. For the recapture PIM, the numbers 7 to 12 correspond to the recapture probabilities p_2 to p_7, respectively. Notice that **MARK** indexes the survival parameters first (1 to 6), followed in numerical sequence by the index values for the recaptures (7 to 12). In other words, the indexing that **MARK** uses does not correspond to the number of the particular interval or occasion involved. For example, the recapture index 8 corresponds to the recapture probability at occasion 3 (i.e., p_3). Although it can be a little confusing at first, with a bit of work you should be able to see the basic connection between the 'true' parameter structure of the model, and the way in which **MARK** indexes it, both internally (which becomes very important later on as we examine more complex models), and in the output file.

We'll be using PIMs frequently throughout the rest of the book, so not to worry if you don't immediately see the connection – you'll get lots of practice. But, make sure you spend some time trying to grasp the connection now. This is <u>very</u> important.

So, getting back to the output:

```
             Real Function Parameters of {phi(t)p(t)}
                                              95% Confidence Interval
    Parameter         Estimate      Standard Error      Lower           Upper
-------------------  ------------   --------------   --------------  --------------
      1:Phi          0.6969698       0.2049829        0.2555234       0.9390713
      2:Phi          0.4230769       0.0968907        0.2519587       0.6148826
      3:Phi          0.5052879       0.0874934        0.3396458       0.6697773
      4:Phi          0.6094017       0.0838412        0.4389147       0.7567907
      5:Phi          0.5708181       0.0776952        0.4166817       0.7123428
      6:Phi          0.7637583     514.12938          0.4489907E-10   1.0000000
      7:p            0.7173912       0.2238535        0.2257359       0.9567135
      8:p            1.0000000       0.5048141E-07    0.9999999       1.0000001
      9:p            0.9093023       0.0855744        0.5674379       0.9871171
     10:p            0.9274193       0.0693437        0.6291492       0.9897162
     11:p            0.9358289       0.0616791        0.6607895       0.9909235
     12:p            0.7637669     514.13517          0.4490122E-10   1.0000000
```

Note that our survival estimates correspond to parameters 1 to 6, while 7 to 12 correspond to the recapture probabilities (corresponding to the figures on the preceding page). But, if you look carefully, you'll notice that the estimates for index 6 (the survival probability from occasion 6 to occasion 7) and index 12 (the recapture probability at occasion 7) are identical (0.76377). As discussed in detail in Lebreton *et al.* (1992), and elsewhere in this book, this reflects the fact that, for this model, the survival and recapture rates for the last interval are not individually identifiable. The value of 0.76377 is actually the square-root of the estimated product $\varphi_6 p_7$ (this is denoted as β_7 in Lebreton *et al.* 1992). Thus, **MARK** has estimated 11 parameters: 5 survival probabilities (φ_1 to φ_5), 5 recapture probabilities (p_2 to p_6), and one related to the product $\varphi_6 p_7$. You may also notice that the SE and 95% CI for one of the estimates ($\hat{p}_3$) is clearly 'suspect' – we'll deal much more with 'problems with the estimates' later.
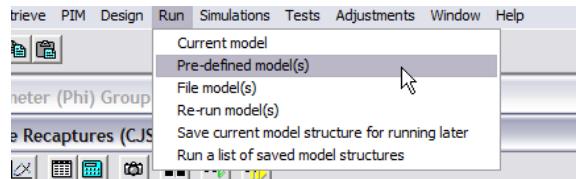
At this point, you can do one of several things. You can print your estimates, you can plot them, or you can examine aspects of the estimation procedure, and look at the degree to which any given capture history in your `.INP` file affects your estimates. We'll defer all the options for the moment, and simply close the notepad window with the estimates, and look back at the main browser window itself.

Much of what **MARK** shows you in the browser is important only in the context of comparing the fit of one model with that of another. In order to demonstrate this, we'll continue our exploration of the dipper data set, by running 3 additional models: $\{\varphi_t\, p_.\}$, $\{\varphi_.\, p_t\}$ and $\{\varphi_.\, p_.\}$, corresponding to time-varying survival and constant recapture ($\{\varphi_t\, p_.\}$), constant survival and time-varying recapture ($\{\varphi_.\, p_t\}$), and constant survival and recapture ($\{\varphi_.\, p_.\}$), respectively. There are **many** more models we could try to fit, but for the purposes of this exercise (which is intended to give you a more complete sense of how the results browser works), we'll run these three.
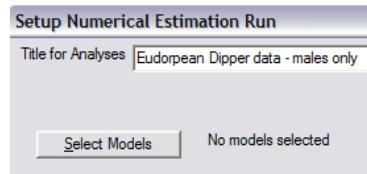
There are a number of ways you could specify these models in **MARK**. For the moment, we'll use a 'short-cut' method which is convenient for some standard models. The short-cut makes use of some

'built-in' models in **MARK**. **Note**: you will be **strongly** discouraged from ever using this short-cut again, but for the moment, it's convenient.
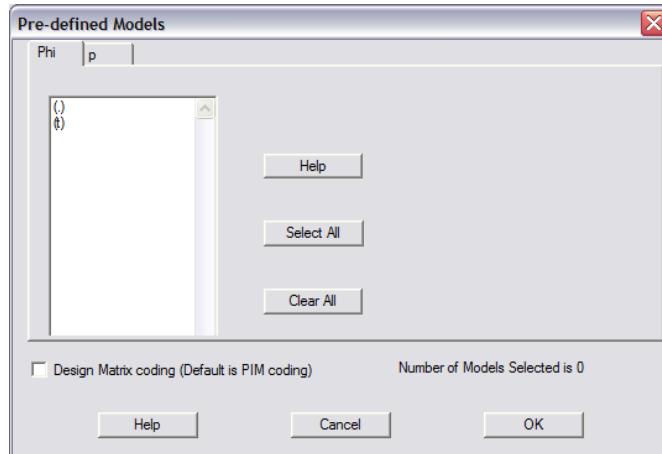
If you pull down the '**Run**' menu, you will see that there are several menu options, beside the 'run the current model' we used earlier. The two options of interest at this point are: '**pre-defined**' models and '**file**' models. For the moment, we're interested only in the pre-defined models.

Once you have selected the '**pre-defined models**' option from the '**Run**' menu, you will be dumped directly back into the '**Run**' window. However, there is one important, but easily overlooked change to the window. Where in the first instance there was a button to fix parameter estimates, note that now this button has been replaced with a '**Select models**' button. Everything else is the same, so you have to be observant. The other difference is that the model title box has been eliminated. In a moment you'll see why. Click on the '**Select models**' button.

Once you've clicked on the appropriate button, you'll be presented with a tabbed-window (shown below), where each tab represents one of the parameters in the models you're working with (in this case, φ and p). All you need to do is (i) select the parameter by clicking the appropriate tab, and (ii) select the model structure(s) you want for each parameter in turn. For example, the following shows the φ parameter – we've selected only the time-invariant 'dot' model {.}. [Don't worry about the 'design matrix' option for the moment – much more on that in Chapter 6.] Selecting both (t) and (.) for both parameters would yield 4 final models: {$\varphi_t p_t$}, {$\varphi_t p_.$}, {$\varphi_. p_t$}, and {$\varphi_. p_.$}. For the moment, we're only interested in the last 3 (since we've already built model {$\varphi_t p_t$}).

So, simply select the {·} option for the φ parameter, as shown above. Note that at this point, **MARK** tells you (in the lower-right hand corner of the window) that the number of models selected to run is 0 – this is because we haven't yet defined the structure for the other parameter p yet. Click the tab for the p parameter, and select both the {*t*} and {·} models. Now, **MARK** reports that you've selected 2 models: {$\varphi_. p_.$} and {$\varphi_. p_t$}. Click the '**OK**' button for the 2 models we've just selected. This brings you back to the '**Set Up Numerical Estimation Run**' window again. Click the '**Run**' button, which will run these 2 models, and automatically add the results to the browser. However, we also want to run model {$\varphi_t p_.$}. Simply go through this process again (i.e., running a pre-defined model), except this time, select {*t*} for the φ parameter, and {.} for the p parameter. Click '**OK**', and run this model, adding the results to the browser.

Note that when running pre-defined models, you do not have the option of fixing parameters, and there is no input box to specify the model title. The reason? Simple – **MARK** figures that if you're using the pre-defined models, you're willing to accept the parameter structure and model names that **MARK** uses for defaults. Of course, **MARK** still allows you to choose among the various link functions, variance estimation routines, and other options normally associated with the run window. If you're satisfied with what is set for these other options, then simply click the '**OK to run**' button in the lower right-hand corner of the run window.

Running pre-defined models in **MARK** is somewhat different from the way they run normally (i.e., the way our initial model ran). First, running pre-defined windows does not cause **MARK** to spawn a command window showing the progress of the numerical estimation. Second, **MARK** does not ask you after each model is finished whether or not you want to add the results to the results browser – **MARK** assumes you do, and sends the results to the browser automatically.

Once the processing of the models is complete, you will note that they have been added to the results browser:
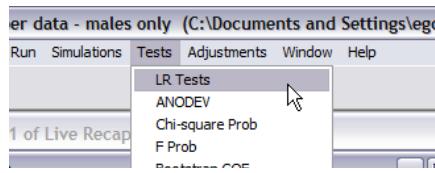
Model	AICc	Delta AICc	AICc Weight	Model Likelihoo	No. Par.	Deviance
{Phi(.) p(.) PIM}	322.5527	0.0000	0.96003	1.0000	2	41.8147
{Phi(t) p(.) PIM}	330.0567	7.5040	0.02253	0.0235	7	38.8147
{Phi(.) p(t) PIM}	330.6794	8.1267	0.01650	0.0172	7	39.4374
{phi(t)p(t)}	336.4343	13.8816	0.00093	0.0010	11	36.4013

However, note that the results are not listed in the order in which the models were processed. In fact, if you'd been watching the results browser while **MARK** was processing the list of models, you might have noticed that the ordering of the models in the list changed with the completion of each model. In fact, **MARK** is ordering (or sorting) the results based on some sort of criterion – in this case, in ascending sequence starting with the model with the lowest AIC_c value (for the dipper example, this corresponds to model '`Phi(.)p(.)`' – constant survival and constant recapture). The sorting criterion can be controlled using the '**Order**' menu.

We will talk a **lot** more about AIC_c in subsequent chapters. For the moment, accept that the AIC_c is a good, well-justified criterion for selecting the most parsimonious model (i.e., the model which best explains the variation in the data while using the fewest parameters). In a very loose sense, we might state that the model with the lowest AIC_c is the 'best' model (although clearly, what is 'best' or 'worst' depends upon the context). The results browser shows you the AIC_c for each model, as well as the arithmetic difference between each model and the top model (i.e., the one with the lowest AIC_c value). For example, model '`Phi(t)p(.)`' has an AIC_c value of 330.06, which is 7.50 units larger than the AIC_c for the most parsimonious model (model '`Phi(.)p(.)`', which has an AIC_c value of 322.55).

The right-most column (by default) is the model deviance. In simple terms, the lower the deviance, the better the model fits. The technical details of the estimation of the likelihood and deviances are given in Lebreton *et al.* (1992). We'll talk more about the deviance (and related statistics) later.

Perhaps more notably, the difference in deviance between 'nested models' (models in which one model differs from another by the elimination of one or more model terms) is distributed as a χ^2 statistic with the degrees of freedom given as the difference in the number of estimable parameters between the two models. This forms the basis of the *likelihood ratio test* (LRT). In fact, **MARK** provides a variety of 'statistical tests' for comparing among models, including the LRT. To perform a LRT on the models in the results browser, simply pull down the '**Tests**' menu, and select '**LR tests**' (as shown below). (*Note*: the relationship between AIC_c, model selection, and 'classical' statistical tests like the LRT will be presented in more detail in subsequent chapters.)

MARK will then present you with a window allowing you to select the models you want to compare using LRT. For now, simply '**Select all**', and then click the '**OK**' button. You will be shown the results of the LRT tests in a Notepad window.

Now, what do we note about the results? Most importantly, we see that not all paired-comparisons among models are possible – the comparison between model $\{\varphi_t \, p.\}$ and model $\{\varphi. \, p_t\}$ is not calculated. Why? If you recall from the preceding page, LRT may be applied only to 'nested' models. We'll talk more about 'nesting' in the next chapter, but for now, accept that these 2 models are not nested. As such, we cannot use an LRT to compare the fit of the 2 models. In one sense (although perhaps not the most appropriate one), this is one of the advantages of using the AIC_c to compare models – it can be used regardless of whether or not the models are nested. [However, as we will discover in later chapters, there may be far more important reasons to use AIC_c as an omnibus (general) model selection tool.]

For the moment, concentrate on the comparison of model $\{\varphi. \, p.\}$ with model $\{\varphi_t \, p.\}$, the first 2 models noted in the editor window (above). We note that the difference in the model deviances is 3, with a difference in the number of parameters of 5. Based on a χ^2 distribution, this difference is not significant at the nominal $\alpha = 0.05$ level ($P = 0.700$). In other words, both models fit the data equally well (we cannot differentiate between them statistically). The model with more parameters fits the data better in terms of the model deviance, but not so much so as to compensate for the fact that it takes more parameters to achieve this better fit. Thus, we would conclude that model $\{\varphi. \, p.\}$ is the more parsimonious model,

which is consistent with the results using the AIC_c.

Of course, at this point you can browse the estimates, plot them, examine residual plots – simply by selecting the model you're interested in (by clicking on it in the results browser). You can also re-run any particular model, using (for example) a different link function, simply by selecting the model from the list, retrieving the model (the '`Retrieve`' menu), and then re-running the model (specifying the new link function, of course). **MARK** will process the data, and then ask you if you want to add the new results to the browser. It's that easy.

Once you're done working with this project, all you need to do is exit **MARK**. All the model results will be stored away in a DBF file (in this case, called **ed_males.dbf**). Then, if you want to continue work on this analysis, all you'll need to do is start **MARK**, and then '`Open`' the **ed_males.dbf** file. That's it!

3.5. Summary

Congratulations! You've just finished your first analysis using program **MARK**. Of course, the fact that there are hundreds (upon hundreds) of pages left in this book should tell you that there is a lot more remaining to be covered. But, you've at least gotten your feet wet, and run through a 'typical' **MARK** analysis once – this is an important first step. If you don't feel comfortable with what we've done so far go back through the chapter – slowly. Many of the basic mechanics and presented in this chapter (in particular, the relationship between model 'structure' and the PIMs) will be used repeatedly throughout the rest of the book, so it is important to feel comfortable with them before proceeding much further.

3.6. References

Lebreton, J.-D., Burnham, K. P., Clobert, J., and Anderson, D. R. (1992) Modeling survival and testing biological hypotheses using marked animals: a unified approach with case studies. *Ecological Monographs*, **62**, 67-118. `doi:10.2307/2937171`