

Exercise 3: The Multistrata Model

This exercise closely follows the content of the 6th lecture and is mostly intended to show how to run program MARK to compute survival, transition and capture probabilities from 'capture-recapture' data.. We have prepared exercises for use in either MARK or RMark.



Data input

Background

Data for this example came from the trapping of meadow voles, *Microtus pennsylvanicus*, at Patuxent Wildlife Research Center, Laurel, MD (Nichols *et al.*, 1984). Data were collected on a 10 x 10 grid of trapping stations spaced at 7.6m intervals in old field habitat. A single modified Fitch live trap (Rose, 1973) was placed at each station. Hay and dried grass were placed in the traps and whole corn was used as bait. Sampling occurred for five consecutive days each month, from June 1981 through December 1981. During each 5-day trapping session, traps were opened in the evening of the first day, checked the following morning, locked open during the day, and reset in the evening, with the sequence repeated each day until 5 days had elapsed. A racoon, *Procyon lotor* (later captured), visited the traps on the final two nights of the second trapping session, essentially leaving only 3 days of trapping for this session. At each capture, animals were examined for a tag, sexed, weighed, and examined for external reproductive characteristics. Tagged animals were eartagged with numbered fingerling tags, and tag numbers of marked animals were recorded at each capture.

We used 'adult' (>22g) animals and collapsed the 5 days of sampling each month into a single assessment of presence or absence, leaving 11 monthly sampling occasions. For each capture, the location of capture was recorded. Individuals captured in locations with X-coordinate in the range 1-5, were assigned capture-code '1'. Those captured in locations with X-coordinate in the range 6-10 were assigned capture-code '2'.

For R users

- 1) Open the file “multistrata.r” in your preferred text editor for R. **You will need to modify the working directory.** Execute the lines to clear the workspace, set the working directory and load the RMark library.

```
rm(list=ls()); setwd('h:/x/workshops/uf2016/exercises/ex3_multistrata_mp')  
library(RMark)
```

- 2) The first step is to convert the input file into an RMark input “data-frame. This conversion is done with the “convert.inp” function. The only required argument is the input filename (mp2age.inp), but since we have 2 groups in our input file, we also need to specify the group names to the convert.inp function. This is done with the group.df argument. The converted data-frame is saved with the name, “mpinp”. Execute that line, then type the new variable name into the R window.

```
mpinp = convert.inp('multistrata.inp',group.df=data.frame(sex=c('F','M')))
```

- 3) The 2nd step is to create a processed-data variable which contains other variables needed to setup and run the MARK models. The RMark function to do this is “process.data” and requires the converted input from the previous step, the type of MARK models which will be run, and the group variable name (if applicable). Execute this line, then type ‘mppr’ to see the contents of this variable.

```
mppr = process.data(mpinp, model='Multistrata', groups="sex")
```

- 4) The 3rd step is to create design-matrix data variables, needed by MARK to build models. The RMARK function is “make.design.data” and the processed-data variable created in the previous step is needed as an argument to the function. Execute this line.

```
mpdd = make.design.data(mppr)
```

Modeling strategy is to develop models corresponding to our hypotheses of how survival, transition and/or capture probabilities are affected by sex and time. One of the primary objectives of the study was to determine the effects of habitat fragmentation. Just after the 4th month, a strip was mowed down the middle of the area. The hypothesis was that the mowing of the strip would cause movement of animals between the two grid halves to decrease after the 4th month.

- 5) Now we're ready to build our first MARK model. We'll start with the most simple model (survival transition and capture probabilities constant over time). To create a model, we call the 'make.mark.model' function with the processed-data variable created in step 3 (mppr), the design-data variable created in step 4 (mpdd), title, and list of parameters as arguments. As a reminder, the formula, "~1" means that the parameter is constant over all values of time and group (sex).

```
s_1_psi_1_p_1 = make.mark.model(mppr,mpdd,title='Patuxent MP data',parameters=list(
  S=list(formula=~1),
  Psi=list(formula=~1),
  p=list(formula=~1)
))
```

- 6) We can run the model by calling the 'run.mark.model' function with the model variable created in step 5 as the argument. Execute this line then type 's_1_psi_1_p_1_out' to examine the contents. The output from MARK is stored in a text file and by typing 's_1_psi_1_p_1_out', the text file is opened in notepad. This is the usual output you would get if you ran MARK interactively using its GUI. The output is also stored as an R list-variable. Type 'str(s_1_psi_1_p_1_out)' to display the structure of this list-variable.

```
s_1_psi_1_p_1_out=run.mark.model(s_1_psi_1_p_1)
```

- 7) To run other models, we only need to repeat the last 2 steps for each model. Run the next model by executing the appropriate lines in the file.

```
s_t_psi_t_p_t = make.mark.model(mppr,mpdd,parameters=list(
  S=list(formula=~time),
  Psi=list(formula=~time),
  p=list(formula=~time)
))
s_t_psi_t_p_t_out = run.mark.model(s_t_psi_t_p_t)
```

- 8) To make a model where transition rates were one value before the mowing of the strip, and another value after the mowing, we'll need to create a new design-data variable. We'll call it "twoper" and set it to zero for the 1st 4 months and one for the last 6 months.

```
mpdd$Psi$twoper=0
mpdd$Psi$twoper[mpdd$Psi$Time>3]=1
```

- 9) We can now build and run our model where transition rates are one value for the 1st 4 months and another value after.

```
s_t_psi_2per_p_t = make.mark.model(mppr,mpdd,parameters=list(
  S=list(formula=~time),
  Psi=list(formula=~twooper),
  p=list(formula=~time)
))
s_t_psi_2per_p_t_out = run.mark.model(s_t_psi_2per_p_t) #      run model
```

- 10) We can create a table of model results by calling the 'model.table' function, with the names of the variables which contain the output of each model as an argument. Execute this line, and the next line to print the table.

```
tbl=model.table(model.list=ls(pattern="s_."+out"),type="Known")
print(tbl)
```

Questions:

- (1) This grid was part of an experiment designed to test hypotheses about effects of fragmentation on meadow vole population dynamics. If the fragmentation created by the strips of bare ground really affected movement, what predictions would we make about effects of fragmentation on the 3 sets of model parameters (survival, capture and movement probabilities)?
- (2) Based on AIC, which model appears to be best supported by the data? What conclusions can you draw from this experiment based on the AIC table? Are the parameter estimates themselves relevant to conclusions or does AIC provide all of the information that you need?
- (3) Have a look at the estimates of movement probabilities from the 2 time periods, before and after fragmentation. Are they consistent with your predictions?
- (4) Do the results from this grid provide strong inferences about effects of fragmentation? If not, what other information would be useful in strengthening the inferences?
- (5) In (1) did you make any predictions about changes in survival probability associated with fragmentation? If so, what was your rationale? The top model provided time-specific estimates of survival, so we could compute and compare means for the periods before and after fragmentation. Another way to obtain inference about this contrast is to look at estimates arising from the 3rd model, in which survival is computed for 2 time periods (before and after fragmentation). What do these look like? Are they consistent with predictions?