Modeling Survival in a Population of Bannertail Kangaroo Rats Using Data from a Portal, Arizona Study Site

Becky Carpenter
Undergraduate in Ecology, Purdue University. Research Professor: Dr. Peter Waser

Abstract

We modeled a population of bannertail kangaroo rats using a program written with dBASE III to look into the effects of immigration on population persistence. The program takes actual data collected at a study site of a possible metapopulation of kangaroo rats in Portal, Arizona, and models a population up to 25 generations into the future. The first part of the program models a population using only survival and recruitment data; the second part includes immigration data. Preliminary results show that immigration seems to increase the number of generations that the modeled population persists for. There is still some concern for the validity of the program, especially the random number generator included with it, however, we feel that with a few changes this program can be a valid test of the effects of immigration on a population.

Introduction

A metapopulation occurs when there are a set of local populations included in one larger area where it is possible for members of one population to immigrate to another (Hanski 1997). It is believed that the migration between these local populations lead to a rescue effect that increases the probability of

survival of the populations when compared to populations that do not see much immigration (Hanski 1997, 2000).

We wanted to build a computer model of a population of bannertail kangaroo rats in Portal, Arizona, to see what role, if any, immigration played in the survival of the population. There are eight small populations in the study area, and they are believed to be a part of one metapopulation. The program, written with dBASE III, accesses the data collected on survival, recruitment, and immigration for this population and uses it to predict population sizes for up to 25 generations of the animals, first without immigration effects on the population and then with immigration effects.

The idea behind this program was twofold: to investigate whether or not the population of kangaroo rats in the Portal study site is actually a metapopulation, as well as to look at the effect immigration can have on population persistence. It is believed that metapopulations are strongly influenced by immigration - the subpopulations within the metapopulation feed each other individuals which increases the populations' persistence over time. We hope to be able to use this program to correctly model the population of kangaroo rats and show what effect immigration may have on it.

Materials and Methods

The data used in this model came from an ongoing study of a metapopulation of kangaroo rats in Portal, Arizona (where there are 8 small populations included in the study area). The model was written using DBASE III. The program (included at the end of this paper) uses the actual survival, recruitment, and immigration data to model a population up to 25 generations. A copy of the file (becky.dbf) with these data is also included at the end of this paper.

The program has two main parts. The first is a random number generator. The idea was to be able to randomly select one of the record numbers in the becky.dbf file and select its survival, recruitment, and immigration values. The second part uses this information to begin generating population sizes for the modeled population. The program does this twice - once without taking immigration into account, and then with immigration factored in.

The equation for the random number generator was taken from *Advanced Turbo Pascal* by Herbert Schildt:

$$R_{n+1} = (aR_n + c) \bmod m$$

where R, a, and c cannot be less than zero and m must be larger than R, a, and c. In this program the user gets to select the initial R value, while a and c have been set at constant values. Schildt suggests that a should be fairly large and c fairly small, so they have been given values of 999 and 2, respectively. The user is asked to select a value for R between 1 and 1000. Since m must be larger than all the other values, it is set to 1001 in the program.

This is the setup for the random number generator. Once the user selects an R value, the program then begins. It firsts generates a new R value, denoted by X in the program. This value is then converted into Y, the record number in becky.dbf that the program is asked to find. It goes to becky.dbf, finds the particular record number that was generated, and then stores the data in that record to output.dbf. (When immigration is taken into account, the data is stored to output2.dbf. Examples of these two files are also included at the end of this paper.) The X value is then saved to R and the cycle begins again, up to 25 generations. It is specified in the program that the population size (N) is not to rise above 30 individuals; should it do so, then the program inputs the value 30 to N and continues. The population size is initially set to 25 individuals. The program also specifies that it should terminate if the population size drops below one.

The program first uses the initial R value to predict population sizes when survival and recruitment rates are taken into account. It then runs a second time, starting with the initially selected R value, this time taking immigration rates into account as well. Overall, the program runs 50 trials per R value entered by the user: at the end of the first trial, it takes the last R value generated and uses that for the first R value of the second trial, and so on. The files output.dbf and output2.dbf are set up to record the trial number, how many generations make it though each trial, the population size at each generation, and the record number from becky.dbf that the survival, recruitment, and immigration rates came from.

Discussion

Results from a sample run are included as output.dbf and output2.dbf with this paper. Only the first two pages of each file have been printed out, both because they are very lengthy, and because the population sizes begin to repeat within the first two pages as well. Preliminary results show that immigration tends to cause a population to persist for a longer period of time. Looking at output.dbf one can see that the population makes it to 22 generations in the first run, 11 in the second, and 6 for all the rest. Ouput2.dbf, which includes immigration, shows that the population makes it to 24 generations for every run.

The largest problem with this program is with its random number generator. Schildt states in his book that it is best to use several generators at once so that repeats are not seen in the numbers generated. Tests of this program have shown that for almost every run there are eventually repeats in the population sizes. It is possible that adding another couple of loops to the program with changes to the generator equation will help to solve this problem. Also, results can greatly differ depending on the *R* value selected by the user. So far the majority of the ones tested show that immigration is a boost to population persistence, but there appears to be some specific ranges of *R* values for which this is not the case. There are also certain ranges of *R* values that result in either very small or very large numbers of generations per run. Knowing these ranges could bias the user's choice of *R* value.

Another problem is with the repeats in generation numbers and population sizes as seen in output.dbf and output2.dbf. Almost all of the trials have resulted in repeats such as these. It has not been determined yet why exactly this happens, but it is presumed that it has to do with the poor quality of the random number generator.

The point of creating this program was both to investigate whether or not the population of kangaroo rats in the Portal study site is actually a metapopulation, as well as to look at the effect immigration can have on population persistence. It is believed that metapopulations are strongly influenced by immigration - the subpopulations within the metapopulation feed each other individuals which increases the populations' persistence over time. The trials run so far indicate that when

immigration is added into a population, it persists longer than without the immigration. This is hopefully evidence that the Portal populations are indeed a metapopulation.

There are several improvements that can be made to this program in the future. One is to improve the random number generator so that the repeats in population sizes are not seen. Another is to create a second program using this one that allows the user to pick which one of the eight Portal populations he or she wishes to model, and then pull the survival, recruitment, and immigration data that is related only to that population. This will allow for comparisons across the eight populations so that it can be determined if they are sources or sinks.

```
*** THIS IS A PROGRAM THAT MODELS POPULATION SIZES WITH AND WITHOUT THE
EFFECTS
*** OF IMMIGRATION.

*** THIS SECTION INCLUDES THE FILES THAT ARE USED BY THIS PROGRAM. BECKY.DBF
*** CONTAINS THE SURVIVAL, RECRUITMENT, AND IMMIGRATION RATES. OUTPUT.DBF
*** IS WHERE THE RESULTS NOT INCLUDING IMMIGRATION DATA ARE SAVED. OUTPUT2.DBF
*** IS WHERE THE RESULTS INCLUDING IMMIGRATION DATA ARE SAVED.


SET TALK OFF
SET DECIMALS TO 3
SELECT A
USE BECKY.DBF
SELECT B
USE OUTPUT.DBF
SELECT C
USE OUTPUT2.DBF


*** HERE THE USER IS ASKED TO INPUT AN R VALUE FOR THE RANDOM NUMBER
GENERATOR.
*** MRUN<51 TELLS THE PROGRAM TO LOOP THROUGH 50 TIMES.

INPUT "SELECT VALUE FROM 1-1000 FOR R: " TO R
SELECT A
STORE 1 TO MRUN
DO WHILE MRUN<51


*** 25 IS THE INITIAL POPULATION SIZE. N AND R ARE STORED TO NIN AND RIN
*** RESPECTIVELY SO THAT THESE INITIAL VALUES WILL BE USED WHEN THE
```

```
*** PROGRAM RUNS WITH THE IMMIGRATION VALUES.

STORE 25 TO N
STORE N TO NIN
STORE R TO RIN


*** A AND C ARE VALUES IN THE RANDOM NUMBER GENERATOR. THESE HAVE BEEN
*** PRESET SO THAT THE USER DOES NOT HAVE TO WORRY ABOUT THEM.

SET DECIMALS TO 3
STORE 999 TO A
STORE 2 TO C


*** THIS TELLS THE PROGRAM TO GO TO OUTPUT.DBF AND CREATE A NEW ENTRY.
*** IT SETS THE INITIAL GENERATION WITH POPULATION SIZE 25 AS ZERO,
*** AND BY REPLACING RUN WITH MRUN THE FILE WILL LIST WHICH RUN GOES
*** WITH EACH GENERATION.

SELECT B
APPEND BLANK
REPLACE GENERATION WITH 0
REPLACE POPSIZE WITH 25
REPLACE RUN WITH MRUN
SELECT A



*** THIS TELLS THE PROGRAM THAT EACH RUN SHOULD HAVE NO MORE THAN 25
*** GENERATIONS.

STORE 1 TO COUNT
DO WHILE COUNT<25


*** THIS IS THE FORM OF THE RANDOM NUMBER GENERATOR EQUATION. IT IS STORED TO
*** B TO MAKE THE NEXT SECTION EASIER TO READ.

   STORE (A*R + C) TO B


*** THIS SAVES THE FINAL RESULT OF THE NUMBER GENERATOR TO X, WHICH WILL
*** BE THE NEXT R VALUE USED BY THE PROGRAM. Y REFERS TO THE RECORD NUMBER
*** IN BECKY.DBF THAT THE PROGRAM IS TRYING TO ACCESS. THIS EQUATION FOR Y
*** MAY CHANGE BASED ON THE NUMBER OF RECORDS IN BECKY.DBF (HERE, 36). THE
*** ONE IS ADDED SO THAT A RECORD NUMBER OF ZERO WILL NOT BE PRODUCED. THE
*** GOTO Y COMMAND TELLS THE PROGRAM TO GO TO THE RECORD NUMBER THAT MATCHES
*** THE Y VALUE GENERATED. IT THEN SAVES SURVIVAL (SURVR) TO S AND RECRUITMENT
*** (RECRUITR) TO T. (LATER IN THE PROGRAM IT ALSO SAVES IMMIGRATION (IMMR)
TO
*** I.)

   X = MOD(B,1001)
   Y = INT((X/1000)*36)+1

   GOTO Y

      STORE SURVR TO S
      STORE RECRUITR TO T
```

*** HERE THE PROGRAM CALLS OUTPUT.DBF AGAIN. THE EQUATION (N*T + N*S) GIVES THE
*** NEW POPULATION WHEN RECRUITMENT AND SURVIVAL ARE TAKEN INTO ACCOUNT. IT
*** SAVES THESE DATA TO OUTPUT.DBF. IT ALSO SAVES WHICH RECORD NUMBER Y WAS USED
*** SO THAT LATER THE USER CAN EXAMINE THE RESULTS TO SEE IF ANY RECORD NUMBER
*** IS FAVORED OVER ANOTHER.

```
      SELECT B
      APPEND BLANK
      REPLACE GENERATION WITH COUNT
      REPLACE POPSIZE WITH ROUND((N*T + N*S), 0)
      REPLACE RUN WITH MRUN
      REPLACE YVALUE WITH Y
      SELECT A
```

*** THIS SAVES THE NEW POPULATION SIZE TO N, SO THAT THE NEXT GENERATION WILL
*** START WITH THAT POPULATION SIZE AND NOT 25 AGAIN.

```
   STORE (N*T + N*S) TO N
```

*** THESE TWO IF STATEMENTS TELL THE PROGRAM TO: FIRST, IF THE POPULATION SHOULD
*** RISE ABOVE 30, THEN TO SAVE 30 AS THE POPULATION SIZE FOR THE NEXT
*** GENERATION. SECOND, THAT IF N SHOULD DROP BELOW ONE, THEN TO PUSH THE COUNT *** ABOVE 25 SO THAT THE PROGRAM JUMPS TO THE NEXT TRIAL.

```
   IF N>30
   STORE 30 TO N
   ENDIF

   IF N<1
   STORE 50 TO COUNT
   ENDIF
```

*** THIS TELLS THE PROGRAM TO REPLACE THE R VALUE THE USER CHOSE WITH THE
*** CALCULATED X VALUE FOR THE NEXT GENERATION. STORING COUNT+1 TO COUNT
*** TELLS THE PROGRAM THAT IT HAS MOVED ON TO THE NEXT GENERATION.

```
   STORE X TO R

   STORE COUNT+1 TO COUNT

ENDDO
```

*** THE REST OF THE PROGRAM IS VERY SIMILAR TO ABOVE. THE DIFFERENCE IS THAT
*** THIS PORTION INCLUDES IMMIGRATION DATA IN THE CALCULATIONS. THESE ARE
*** SAVED TO OUTPUT2.DBF. NIN AND RIN ARE USED SO THAT THESE CALCULATIONS
*** CAN START OUT WITH THE SAME R AND N VALUES THAT THE FIRST PART OF THE

```
*** PROGRAM STARTED WITH.


SELECT C
APPEND BLANK
REPLACE GENERATION WITH 0
REPLACE POPSIZE WITH 25
REPLACE RUN WITH MRUN
SELECT A

STORE 1 TO COUNT
DO WHILE COUNT<25

STORE (A*R + C) TO B

X = MOD(B,1001)
Y = INT((X/1000)*36) + 1

GOTO Y

STORE SURVR TO S
STORE RECRUITR TO T
STORE IMMR TO I

*? ROUND((NIN*T + NIN*S + NIN*I), 0)

SELECT C
APPEND BLANK
REPLACE GENERATION WITH COUNT
REPLACE POPSIZE WITH ROUND((NIN*T + NIN*S + NIN*I), 0)
REPLACE RUN WITH MRUN
REPLACE YVALUE WITH Y
SELECT A

STORE (NIN*T + NIN*S + NIN*I) TO NIN

   IF NIN>30
   STORE 30 TO NIN
   ENDIF

   IF NIN<1
   STORE 50 TO COUNT
   ENDIF


STORE X TO R
STORE COUNT+1 TO COUNT

ENDDO

STORE MRUN+1 TO MRUN
ENDDO
```

BECKY.DBF

AREA	RECRUITR	IMMR	SURVR
R1E	0.20	0.40	0.82
R1E	0.18	0.36	0.47
R1E	0.60	0.20	0.47
R1E	0.84	0.26	0.47
R1E	0.43	0.20	0.48
R1E	0.52	0.06	0.40
R1E	0.51	0.14	0.46
R1E	0.46	0.11	0.37
R1W	0.33	0.33	0.62
R1W	0.50	1.00	0.71
R1W	0.76	0.06	0.69
R1W	0.31	0.08	0.39
R1W	0.46	0.14	0.57
R1W	0.50	0.00	0.53
R1W	0.37	0.13	0.42
R1W	0.35	0.06	0.38
R2	0.21	0.07	0.50
R2	0.27	0.23	0.47
R2	0.43	0.33	0.54
R2	0.19	0.41	0.29
R2	0.36	0.10	0.23
R3	0.40	0.80	0.38
R3	0.25	0.62	0.50
R3	0.20	0.50	0.33
R3	0.50	0.08	0.27
R4	0.17	1.00	0.22
R5	0.00	1.00	0.14
R5	0.14	1.14	0.60
R5	0.60	1.20	0.67
R5	0.38	0.75	0.40
R5	0.16	0.23	0.53
SSW	0.20	0.40	0.75
SSW	0.50	0.25	0.50
SSW	1.00	0.33	0.36
SSE	0.67	0.67	0.17
SSE	1.33	0.17	0.70

OUTPUT.DBF

RUN	GENERATION	POPSIZE	YVALUE
1	0	25	
1	1	34	42
1	2	29	8
1	3	10	30
1	4	14	32
1	5	7	28
1	6	5	35
1	7	4	21
1	8	4	4
1	9	3	38
1	10	3	15
1	11	3	16
1	12	3	15
1	13	3	17
1	14	3	12
1	15	3	22
1	16	2	3
1	17	2	41
1	18	2	11
1	19	3	25
1	20	3	41
1	21	2	10
1	22	1	28
2	0	25	
2	1	25	41
2	2	24	11
2	3	33	25
2	4	30	41
2	5	20	10
2	6	11	28
2	7	14	36
2	8	6	19
2	9	5	9
2	10	4	29
2	11	1	34
3	0	25	
3	1	13	28
3	2	17	36
3	3	7	19
3	4	6	9
3	5	5	29
3	6	1	34
4	0	25	
4	1	13	28
4	2	17	36
4	3	7	19
4	4	6	9
4	5	5	29
4	6	1	34
5	0	25	
5	1	13	28
5	2	17	36
5	3	7	19
5	4	6	9
5	5	5	29
5	6	1	34
6	0	25	
6	1	13	28
6	2	17	36

6	3	7	19
6	4	6	9
6	5	5	29
6	6	1	34
7	0	25	
7	1	13	28
7	2	17	36
7	3	7	19
7	4	6	9
7	5	5	29
7	6	1	34
8	0	25	
8	1	13	28
8	2	17	36
8	3	7	19
8	4	6	9
8	5	5	29
8	6	1	34
9	0	25	
9	1	13	28
9	2	17	36
9	3	7	19
9	4	6	9
9	5	5	29
9	6	1	34
10	0	25	
10	1	13	28
10	2	17	36
10	3	7	19
10	4	6	9
10	5	5	29
10	6	1	34
11	0	25	
11	1	13	28
11	2	17	36
11	3	7	19
11	4	6	9
11	5	5	29
11	6	1	34
12	0	25	
12	1	13	28
12	2	17	36
12	3	7	19
12	4	6	9

OUTPUT2.DBF

RUN	GENERATION	POPSIZE	YVALUE
1	0	25	
1	1	62	36
1	2	13	19
1	3	12	9
1	4	10	29
1	5	12	34
1	6	8	24
1	7	18	45
1	8	25	2
1	9	43	42
1	10	33	8
1	11	10	30
1	12	20	32
1	13	20	28
1	14	39	35

1	15	29	21
1	16	37	4
1	17	28	38
1	18	32	15
1	19	31	16
1	20	35	15
1	21	28	17
1	22	61	12
1	23	39	22
1	24	30	3
2	0	25	
2	1	17	24
2	2	38	45
2	3	43	2
2	4	51	42
2	5	33	8
2	6	10	30
2	7	20	32
2	8	20	28
2	9	39	35
2	10	29	21
2	11	37	4
2	12	28	38
2	13	32	15
2	14	31	16
2	15	35	15
2	16	28	17
2	17	61	12
2	18	39	22
2	19	30	3
2	20	38	41
2	21	38	11
2	22	162	25
2	23	38	41
2	24	20	10
3	0	25	
3	1	17	24
3	2	38	45
3	3	43	2
3	4	51	42
3	5	33	8
3	6	10	30
3	7	20	32
3	8	20	28
3	9	39	35
3	10	29	21
3	11	37	4
3	12	28	38
3	13	32	15
3	14	31	16
3	15	35	15
3	16	28	17
3	17	61	12
3	18	39	22
3	19	30	3
3	20	38	41
3	21	38	11
3	22	162	25
3	23	38	41
3	24	20	10
4	0	25	
4	1	17	24
4	2	38	45
4	3	43	2

4	4	51	42
4	5	33	8
4	6	10	30
4	7	20	32
4	8	20	28
4	9	39	35
4	10	29	21
4	11	37	4
4	12	28	38
4	13	32	15
4	14	31	16
4	15	35	15
4	16	28	17
4	17	61	12
4	18	39	22
4	19	30	3
4	20	38	41
4	21	38	11
4	22	162	25
4	23	38	41
4	24	20	10
5	0	25	
5	1	17	24
5	2	38	45

Literature Cited

Hanksi, Ilkka, Juha Alho, and Atte Moilanen. 2000. Estimating the Parameters of Survival and Migration of Individuals in Metapopulations. Ecology 81(1): 239-251.

Hanski, Ilkka and Michael E. Gilpin, editors. 1997. Metapopulation Biology: Ecology, Genetics, and Evolution. Academic Press, New York, New York, USA.

Schildt, Herbert. 1986. Advanced Turbo Pascal: Programming and Techniques. McGraw-Hill, Berkeley, California, USA.

Simpson, Alan. 1984. Understanding dBASE II. Sybex Inc., Berkeley, California, USA.