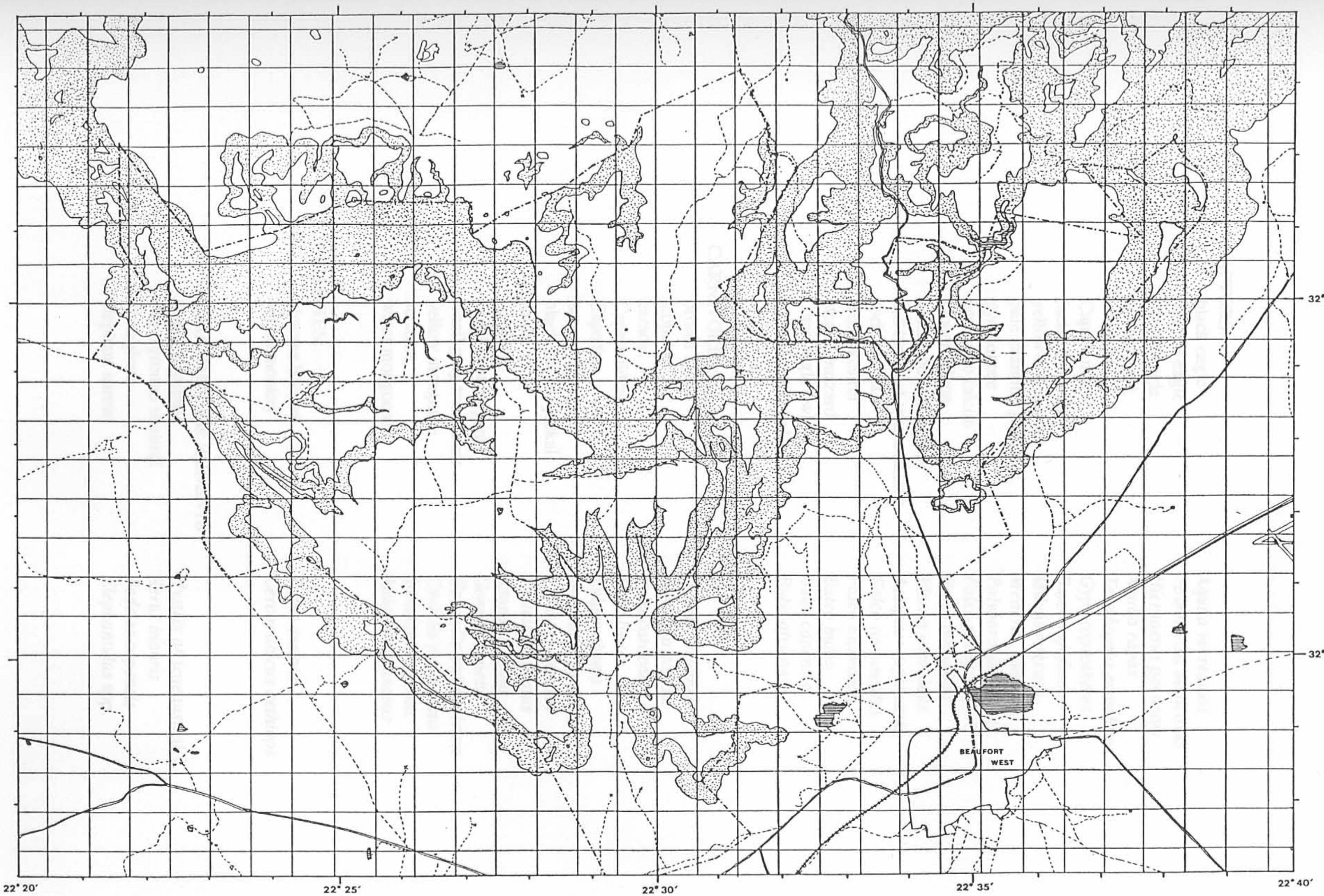
APPENDIX 1

DETAILED MAP OF THE KRNP AND ENVIRONS



LEGEND: stippled areas represent mountain slopes as opposed to plains or plateaus; park boundary is shown in alternating dots and dashes; dashed lines represent dirt roads; double lines represent tarmac roads; dams are darkly shaded in horizontal lines; the grid is one kilometer square, and is the same as that used by previous researchers in the park (Branch, Braack, Randall etc.): grid columns are numbered from right to left commencing with the second intact square from the far right (=#1); grid rows are lettered from A (lowest row, partly shown) to X (uppermost row, partly shown).

APPENDIX 2

SPECIES NAMES OF ANIMALS REFERRED TO IN THE THESIS

RAPTORS:

black eagle	*Aquila verreauxii*
martial eagle	*Polemaetus bellicosus*
booted eagle	*Hieraaetus pennatus*
tawny eagle	*Aquila rapax*
bateleur	*Terathopius ecaudatus*
Cape vulture	*Gyps coprotheres*
jackal buzzard	*Buteo rufofuscus*
yellow-billed kite	*Milvus migrans*
pale chanting goshawk	*Melierax canorus*
gymnogene	*Polyboroides typus*
peregrine falcon	*Falco peregrinus*
lanner falcon	*Falco biarmicus*
black-shouldered kite	*Elanus caeruleus*
red-breasted sparrowhawk	*Accipiter rufiventris*
rock kestrel	*Falco tinnunculus*
lesser kestrel	*Falco naumanni*
steppe buzzard	*Buteo buteo*
Cape eagle owl	*Bubo capensis*
spotted eagle owl	*Bubo africanus*

CARNIVORES:

brown hyena	*Hyaena brunnea*
leopard	*Panthera pardus*
caracal	*Felis caracal*
Cape wildcat	*Felis lybica*
Cape fox	*Vulpes chama*
bat-eared fox	*Otocyon megalotis*
black-backed jackal	*Canis mesomelas*
aardwolf	*Proteles cristatus*
zorilla	*Ictonyx striatus*
small-spotted genet	*Genetta genetta*
small grey mongoose	*Galerella pulverulenta*
yellow mongoose	*Cynictis penicillata*
suricate	*Suricatta suricata*
water mongoose	*Atilax paludinosus*

PRIMATES:

chacma baboon	*Papio ursinus*
vervet monkey	*Cercopithicus aethiops*

SMALL MAMMALS & RODENTS

Cape porcupine	*Hystrix africaeaustralis*
Cape ground squirrel	*Xerus inauris*
springhare	*Pedetes capensis*
elephant shrew	*Elephantulus* spp.

APPENDIX 2 cont.

HYRAX:	
rock hyrax	*Procavia capensis*
LAGOMORPHS:	
Smith's red rock rabbit	*Pronolagus rupestris*
Cape hare	*Lepus capensis*
scrub hare	*Lepus saxatilis*
UNGULATES:	
domestic sheep	*Ovis aries*
domestic goat	*Capra hircus*
klipspringer	*Oreotragus oreotragus*
springbok	*Antidorcas marsupialis*
mountain reedbuck	*Redunca fulvorufula*
grey rhebok	*Pelea capreolus*
steenbok	*Raphicerus campestris*
grey duiker	*Silvicapra grimmia*
BIRDS:	
Ludwig's bustard	*Neotis ludwigii*
karoo horhaan	*Eupodotis vigorsii*
helmeted guinea fowl	*Numida meleagris*
greywing francolin	*Francolinus africanus*
Egyptian goose	*Alopochen aegyptiacus*
African black duck	*Anas sparsa*
yellow-billed duck	*Anas undulata*
redeyed dove	*Streptopelia semitorquata*
rock pigeon	*Columba guinea*
white-necked raven	*Corvus albicollis*
black-necked heron	*Ardea melanocephala*
black stork	*Ciconia nigra*
Alpine swift	*Apus melba*
REPTILES:	
rock leguaan	*Varanus exanthematicus albigularis*
southern rock agama	*Agama atra*
Cape cobra	*Naja nivea*
puff adder	*Bitis arietans*
tent tortoise	*Psammobates tentorius*
greater padloper	*Homopus femoralis*
leopard tortoise	*Geochelone pardalis*

APPENDIX 3

BREEDING HISTORIES OF BLACK EAGLE PAIRS IN THE KRNP

		ANNUAL BREEDING STATUS					
PAIR		1986	1987	1988	1989	1990	1991
KF	Kookfontein		S	S	N	S	
DH	Doringhoek	S	F1	F1	N	N	(N)
SL	Sloop	S	N	S	S	N	(N)
BK	Bakenkop	S	S	F2	S	F3	(A)
LH	Lion's Head	N/F	F4	S	S	S	(A)
BR	Beagle Ravine	F5	S	N	N	N	(N)
KK	Kortkloof	S	N	F6	S	F5	(A)
PB	Penberi	S	N	S	S	S	(A)
ZZ	Zig-zag Kloof	S	F7	S	N	S	(A)
LF	Lemoenfontein		N	F8	S	S	(A)
TG	The Gate			?	?		
GD	Gamka Dam	N/F	S	S	F9	S	(A)
MP?	Molteno Pass	N/F	N	N	N	N	
MV	Mountain View	?		F2	S	S	
TC	Tower Cliffs	S	N	N	N	S	
RC	Red Castle	N	N	N	F3	N	
LC	Lancelot	?	S	N	N	S	
SK	Skomer	?	N	S	S	N	
WC	Weeping Cliffs		?	N	N	N	
VC	Vulture Cliffs		?	S	N	N	
PK	Paardekraal			S	N		
CC	Corner Cliffs			S	N		
LX	Loxton Road	S	S	S	S	S	(A)

KEY: blank spaces represent unmonitored nests; N=not breeding; F=failed attempt; S=successful attempt; A=attempt; ? see notes.

NEST FAILURES:
F1 Galerella pulverulenta predates newly hatched chick
F2 disappearance of small chick
F3 disappearance of large chick
F4 large chick (78d old) dies of impacted crop
F5 unknown
F6 infertile egg
F7 small chick (10d old) abandoned
F8 disappearance of egg or small chick
F9 one egg disappears; remaining small chick (2d old) blown out of nest bowl

NOTES:
MV and LC probably raised chicks, and SK probably didnt breed in 1986 (insufficient data); F7 was caused by the death of resident adult male in August 1987 (drowned in resevoir), ZZ bred again 9 months later in 1988; SL nest (1 of 2) collapsed in 1987 perhaps leading to a non-breeding season; VC probably didnt breed, and WC eggs may have been lost to Corvus albicollis in 1987; juvenile found electocuted near TC eyrie in June 1987 (could have been 1986 fledgling from this site); GD female did not desert despite massive human interference (road building) during the 1988 nesting season; An adult female (possibly TC) was killed by colliding with an FM mast in July 1988, TC bred again 23 months later in 1990; TG probably raised a chick in 1988, and probably made a breeding attempt in 1989 (insufficient data); LF birds may have been shot at on their eyrie during 1989 but they raised a chick nonetheless; WC female (identifiable by white spots) was injured some time after July 1989, she displaced the DH female in 1990, but the WC male was still single for this breeding season; ZZ nest (1 of 2) collapsed in 1991; BR female replaced by new subadult in 1991; Molteno Pass pair never bred and were rarely present in territory. ME in Figure 41 refers to martial eagle.

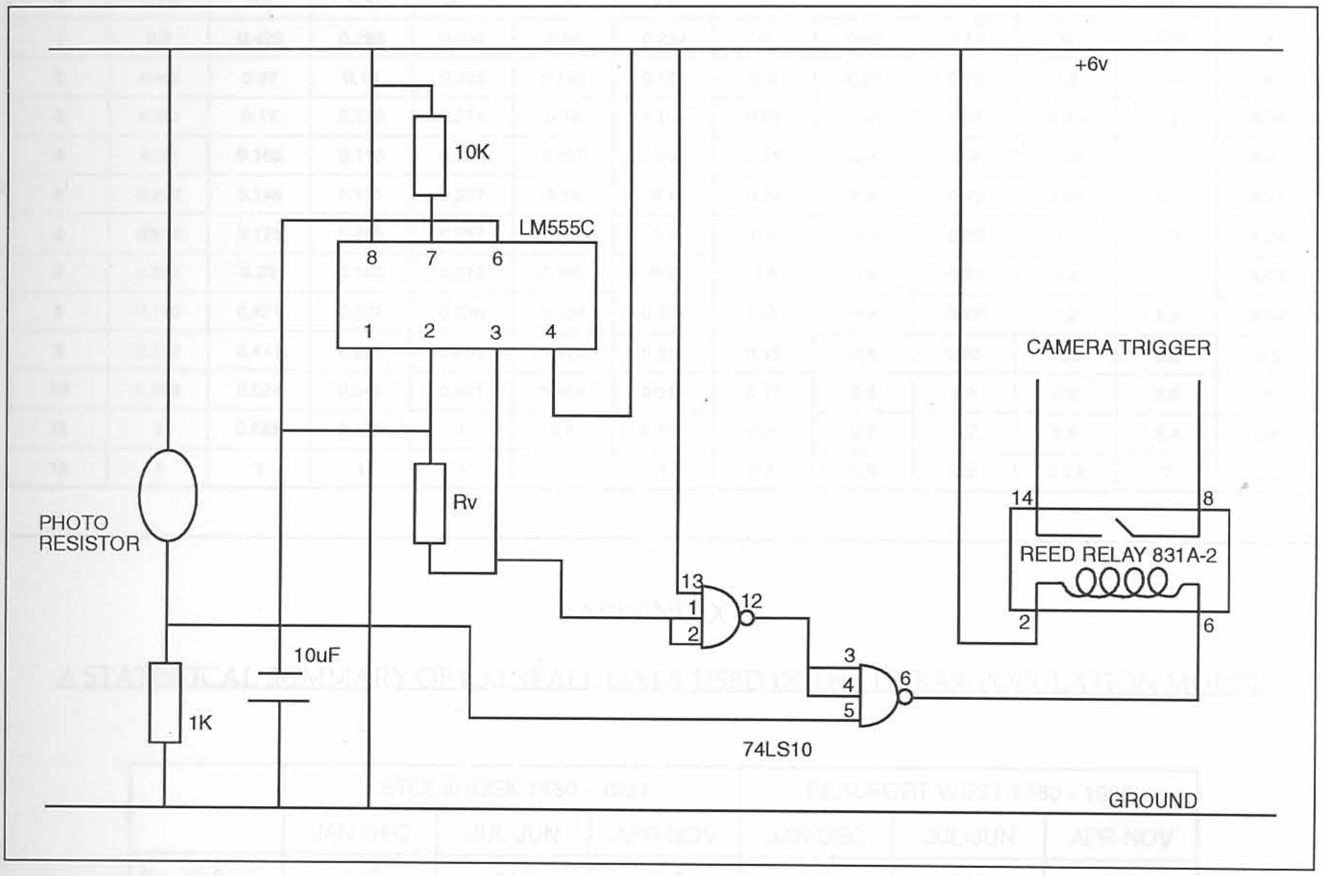
APPENDIX 4
DETAILS RECORDED FOR EACH CARACAL TRAPPED

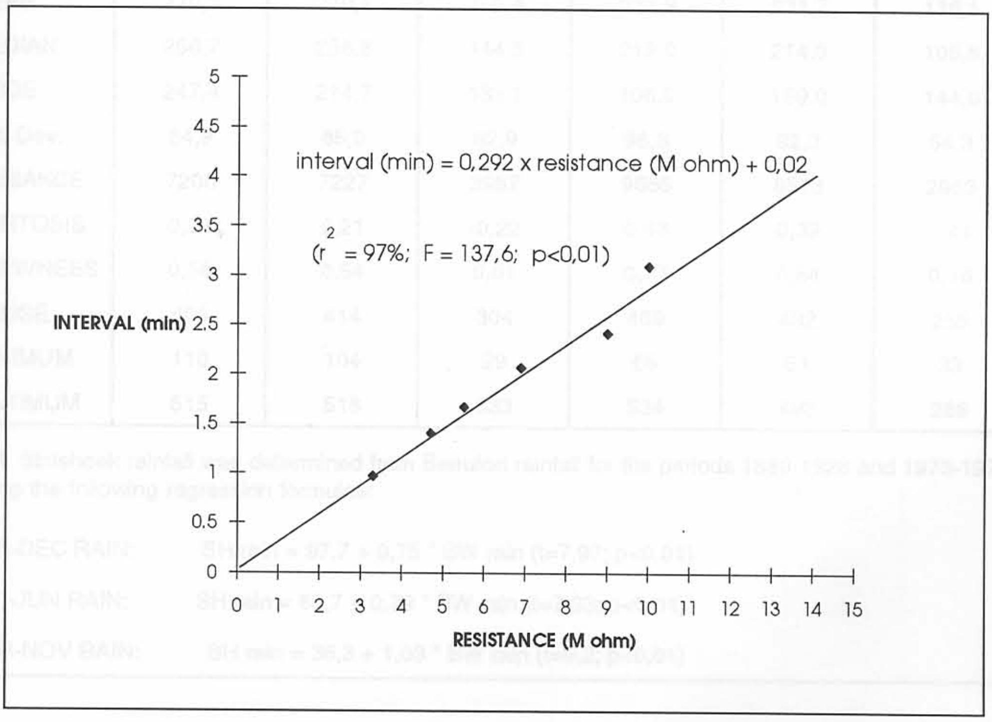
	TANJA	ANDY	MLILO	LUKE
DATE	25/1/88	10/4/88	27/4/88	11/5/88
LOCALITY	Moordenaarsgat	south of Stolshoek	middle plateau	middle plateau
BAIT	none	blood, sand	blood, sand	blood, sand
SEX	female	female	female	male
MASS	4,9kg	5,89kg	10,5kg	14,8kg
CONDITION	fair	good	fair	fair
AGE?	sub-adult	sub-adult	adult	adult (4y)
TOTAL LENGTH	85cm	88cm	111cm	124cm
BODY LENGTH	65cm	64cm	86cm	91cm
SHOULDER HEIGHT		34cm	40cm	44cm
HEAD CIRCUMFERENCE	23cm	24cm	28cm	35cm
NECK CIRCUMFERENCE	18cm	18cm	22cm	26cm
COLLAR SIZE	25cm	23cm	26cm	29cm
ROMPUN	none	0,8ml	1,0ml	1,5ml
KETAMINE	125mg	50mg	100mg	75mg
PERIOD OF ANAESTHESIA	2h	2h	1h 50min	2h 20min
RADIO-COLLAR	B2C9	B1C7	B1C9	B1C5

APPENDIX 5

CIRCUIT DIAGRAM FOR THE TIME-LAPSE SYSTEM AND A PLOT OF THE REGRESSION OF TIME INTERVAL ON VARIABLE RESISTANCE

The circuit was compiled by K. Musgrave (Diginet Systems, Pretoria) after Temple (1972). Part numbers are provided. Rv denotes the variable resistor that could be altered to adjust the time interval (see regression below).





APPENDIX 6

AGE-SPECIFIC FECUNDITY AND MORTALITY DATA USED IN THE HYRAX POPULATION MODEL

AGE CLASS	MALE MORTALITY RATES			FEMALE MORTALITY RATES			PREGNANCY RATES			LITTER SIZES		
	BAD	AVERAGE	GOOD	BAD	AVERAGE	GOOD	BAD	AVERAGE	GOOD	BAD	AVERAGE	GOOD
0	0.75	0.6	0.4	0.75	0.6	0.4	0	0	0	0	0	0
1	0.7	0.425	0.283	0.616	0.36	0.239	0	0.02	0.14	0	0.75	2
2	0.46	0.27	0.18	0.325	0.193	0.126	0.6	0.65	0.75	1.6	1.8	4
3	0.29	0.17	0.113	0.274	0.16	0.106	0.65	0.7	0.85	1.48	2.2	4.24
4	0.29	0.165	0.113	0.274	0.155	0.106	0.75	0.8	0.9	1.28	2.25	4.22
5	0.257	0.145	0.171	0.257	0.15	0.1	0.75	0.8	0.95	1.28	2.5	4.21
6	0.308	0.175	0.205	0.257	0.15	0.1	0.8	0.9	0.99	1.2	2.75	4.04
7	0.393	0.23	0.153	0.513	0.295	0.2	0.8	0.9	0.99	1.2	3.2	4.04
8	0.718	0.421	0.279	0.838	0.494	0.326	0.8	0.9	0.99	1.2	3.2	3.64
9	0.752	0.441	0.293	0.804	0.473	0.313	0.75	0.8	0.95	1.28	3.2	3.3
10	0.889	0.524	0.346	0.821	0.484	0.319	0.75	0.8	0.8	0.8	2.5	3
11	1	0.685	0.452	1	0.8	0.532	0.6	0.7	0.7	0.8	2.4	2.6
12	1	1	1	1	1	1	0.5	0.6	0.6	0.24	2	2.5

APPENDIX 7

A STATISTICAL SUMMARY OF RAINFALL DATA USED IN THE HYRAX POPULATION MODEL

	STOLSHOEK 1880 - 1991			BEAUFORT WEST 1880 - 1989		
	JAN-DEC	JUL-JUN	APR-NOV	JAN-DEC	JUL-JUN	APR-NOV
No. YEARS	112	112	112	110	110	110
MEAN	270,3	270,2	155,4	232,9	231,7	116,1
MEDIAN	256,7	258,8	144,5	212,0	214,0	105,5
MODE	247,4	214,7	133,1	106,0	159,0	144,0
Std. Dev.	84,9	85,0	62,9	98,3	92,3	54,3
VARIANCE	7206	7227	3957	9665	8513	2953
KURTOSIS	0,33	0,21	-0,22	0,43	0,39	0,41
SKEWNESS	0,58	0,54	0,61	0,77	0,84	0,76
RANGE	405	414	304	469	432	255
MINIMUM	110	104	29	65	61	33
MAXIMUM	515	518	333	534	493	288

N.B. Stolshoek rainfall was determined from Beaufort rainfall for the periods 1880-1928 and 1973-1977 using the following regression formulae:

JAN-DEC RAIN: SH rain = 97,7 + 0,75 * BW rain (t=7,97; p<0,01)

JUL-JUN RAIN: SH rain = 89,7 + 0,79 * BW rain (t=7,33; p<0,01)

APR-NOV RAIN: SH rain = 36,3 + 1,03 * BW rain (t=9,2; p<0,01)

APPENDIX 8

PREFERENCE RATINGS FOR SELECTION OF HYRAX AGE-CLASSES BY BLACK EAGLES

These were calculated as %use/%availability after Petrides (1975) - see Chapter 10

AGE CLASS	PREFERENCE RATINGS	
	MALES	FEMALES
0	0.393	0.392
1	1.801	1.297
2	1.175	0.907
3	0.941	1.072
4	1.134	0.916
5	0.817	0.913
6	1.082	0.922
7	1.447	2.035
8	1.763	3.309
9	1.453	3.000
10	2.972	2.191
11	2.732	4.869
12	3.932	4.068

APPENDIX 9

AGE-SPECIFIC DATA ON 'NON-EAGLE' MORTALITY USED IN THE HYRAX POPULATION MODEL

AGE CLASS	MORTALITY RATES TO FACTORS OTHER THAN EAGLES					
	MALES			FEMALES		
	BAD	AVERAGE	GOOD	BAD	AVERAGE	GOOD
0	0.701	0.561	0.374	0.700	0.561	0.374
1	0.383	0.224	0.149	0.355	0.216	0.144
2	0.234	0.139	0.091	0.156	0.092	0.061
3	0.11	0.064	0.042	0.066	0.039	0.026
4	0.065	0.037	0.025	0.091	0.052	0.036
5	0.087	0.051	0.034	0.083	0.047	0.047
6	0.087	0.051	0.034	0.081	0.046	0.046
7	0.117	0.067	0.045	0.113	0.066	0.044
8	0.38	0.224	0.148	0.208	0.112	0.081
9	0.476	0.28	0.185	0.232	0.136	0.084
10	0.322	0.19	0.125	0.405	0.239	0.158
11	0.475	0.38	0.225	0.369	0.253	0.167
12	0.551	0.551	0.551	0.518	0.518	0.518

APPENDIX 10

HYRAX POPULATION MODEL (TURBO PASCALL version 7.0 Borland International Inc.)

{all comments italicised and enclosed in {} are not read by the program}

```
program hyrax(input,output);
USES CRT;
CONST   FETAL_RESORPTION = 0.11;                 {foetal resorption rate}
        RESORPTION_GOODyr = 0.02;                {same for good years}
        RESORPTION_BADyr = 0.2;                  {same for bad years}
        WINTER_CAPITAL = 126994;                 {14.7% of winter capital under standard conditions}
        CAPITAL_CONSTRAINT = 0.3;                {proportion of capital which can be depleted or gained}
        INTEREST_RATE = 0.37267;                 {fresh growth during standard year, as the proportion of standard capital}
        HYRAX_FOOD_CONSUMPTION = 26.061;         {Kg/hyrax/yr}
        STD_KC = 1816;                           {K under standard conditions}
        EDIBILITY = 0.1;                         {proportion of winter capital edible}
        SHEEP_EQUIV = 10;                        {metabolic equivalent of one adult sheep in hyrax}
        COST_PER_SHEEP = 30;                     {cost of letting veld per sheep per year in Rands}

TYPE    age_array = ARRAY[0..12] of REAL;
        year_array = ARRAY[1870..1995] of INTEGER;

VAR     M_prev,M_next,F_prev,F_next,Fecundity,Fec_bd, Fec_av,Fec_gd:                    age_array;
        incmort_F,incmort_F_bd,incmort_F_av,incmort_F_gd,othmort_F:                      age_array;
        othmort_F_bd,othmort_F_av,othmort_F_gd,incmort_M,incmort_M_bd:                   age_array;
        incmort_M_av,incmort_M_gd,othmort_M,othmort_M_bd,othmort_M_av:                   age_array;
        othmort_M_gd,actmort_F,actmort_M,actdie_F,actdie_M,incdie_F:                     age_array;
        incdie_M,othdie_F,othdie_M,Pregnancy,Preg_bd,Preg_av,Preg_gd:                    age_array;
        LitterSize,Litr_bd,Litr_av,Litr_gd,F_die, M_die, Pref_M, Pref_F:                 age_array;
        F_tmp,M_tmp,M_eagle_kill,F_eagle_kill,M_lynx_kill,F_lynx_kill:                   age_array;
        summer_rain,gestn_rain,gestn_rain_SH,summer_rain_SH:                             year_array;
        gestn_rain_BW, summer_rain_BW:                                                   year_array;
        mean_summer_rain, min_summer_rain, max_summer_rain:                              REAL;
        mean_gestn_rain, min_gestn_rain, max_gestn_rain:                                 REAL;
        mean_gestn_rain_SH,mean_gestn_rain_BW,max_gestn_rain_SH:                         REAL;
        max_gestn_rain_BW,min_gestn_rain_SH,min_gestn_rain_BW:                           REAL;
        mean_summer_rain_SH,mean_summer_rain_BW,max_summer_rain_SH:                      REAL;
        max_summer_rain_BW,min_summer_rain_SH,min_summer_rain_BW:                        REAL;
        No_Hyrax_prev,No_Hyrax_next, r, No_males, No_Females, No_babies :                REAL;
        K_prev,SR,fjuv,fimm,f2yr,fads,fold,KC,std_resource,std_growth:                   REAL;
        resorption,K_ratio,P_ratio,K_factor,crash_level,crash_mort:                      REAL;
        capital,food_eaten,food_resource,fresh_growth:                                   REAL;
        fresh_growth_1,prop_growth_eaten,Prod_mult,foet_mult:                            REAL;
        min_mult,max_mult,unpal_buffer,start_pop,sheep,cost:                             REAL;
        correction,Sum_C,prey_availability,eagle_kill,Comp_Old:                          REAL;
        R_diff,total_rain_SH,total_rain_BW,impede,version:                                REAL;
        alt_demand,eagle_alt,lynx_alt,Diff_K,No_HP,Surplus,Lynx_increase:                REAL;
        Lynx_Decline,Lynx_Diet_Min,Lynx_Diet_Max,Lynx_Consumption:                       REAL;
        Dassie_Meat,lynx_mult,Lynx_Nos,Lynx_Prev,Lynx_Food,Surplus_Per_Lynx:             REAL;
        Lynx_Diet,lynx_kill,total_dying,other_dying,pop_growth,juv_loss:                 REAL;
        rec_loss,loss_above,loss_below,mort_above,mort_below:                            REAL;
        no_crashes:                                                                      INTEGER;
        year,i,y,generation,first_date,last_date,start_date,end_date:                    INTEGER;
        rain_locality,yn:                                                                CHAR;
        remove_eagle_effect,age_selection,terminate_model,bias:                          BOOLEAN;
        Demogr_file,Rain_file,outmain_out,outagest_out,outagepr_out:                     TEXT;

procedure eagle_predation; forward;
procedure lynx_predation; forward;

{ *****************************************************************************************************
  ************************************* P R O C E D U R E S ********************************************
  ***************************************************************************************************** }

procedure read_rainfall;
{
==========================================================================================================
        RAINFALL SUB-ROUTINE
        Read in the rainfall data (1878-1989): calculate maximum, minimum and mean values; initialise gestn_rain and summer_rain variables.
==========================================================================================================
}
VAR Rain_file:TEXT;
  no_years,y:INTEGER;
BEGIN
Assign(Rain_file,rain_locality+'_RAIN.DAT');
reset(Rain_file);
mean_gestn_rain := 0;                                      {Initialise accumulators, maxima & minima}
mean_summer_rain := 0;
Min_gestn_rain := 10000;
Max_gestn_rain := 0;
Max_summer_rain := 0;
Min_summer_rain := 10000;
no_years := 0;
readln(Rain_file);                                         {Read file heading with captions}
read(Rain_file,year,summer_rain[year],gestn_rain[year]);
first_date := year;                                        {Determine 1st rainfall year}
repeat
        read(Rain_file,year,summer_rain[year],gestn_rain[year]);       {Read rainfall figures}
        if gestn_rain[year] < Min_gestn_rain then Min_gestn_rain := gestn_rain[year];
```

```
        if gestn_rain[year] > Max_gestn_rain then Max_gestn_rain := gestn_rain[year];        {calculating maxima and minima .... }
        if summer_rain[year] < Min_summer_rain then Min_summer_rain := summer_rain[year];
        if summer_rain[year] > Max_summer_rain then Max_summer_rain := summer_rain[year];
        mean_gestn_rain := mean_gestn_rain + gestn_rain[year];
        mean_summer_rain := mean_summer_rain + summer_rain[year];
        inc(no_years);
until eof(Rain_file)=TRUE;
last_date := year;                                                { Determine last rainfall year }
mean_gestn_rain := mean_gestn_rain/no_years;                      { Calculate mean values }
mean_summer_rain := mean_summer_rain/no_years;
close(Rain_file);
END;

procedure vegetation_grow(Year:INTEGER; No_Dassies: REAL);
{
=====================================================================================================
                    LOWER  TROPHIC  (VEGETATION)  LEVEL
        This procedure takes winter phytomass from the previous year and calculates fresh growth in accordance with summer rainfall
        phytomass subsequently accumulates or slightly declines in accordance with rainfall
        If fresh growth is not sufficient to meet hyrax food demands, the deficit is removed from the phytomass.
=====================================================================================================
}
VAR consumption_ratio,rain_mult:REAL;
BEGIN
 rain_mult := min_mult + (max_mult-min_mult) * (summer_rain[Year-1]-min_summer_rain)/(max_summer_rain-min_summer_rain);
 capital := capital * rain_mult;                                  {modify phytomass according to rain}
 food_eaten := No_Dassies * DASSIE_FOOD_CONSUMPTION;
 consumption_ratio := food_eaten/fresh_growth;
 if consumption_ratio > 1 then
   capital := capital - (1-unpal_buffer)*(food_eaten - fresh_growth);   {depletion of phytomass by heavy grazing}
 if capital > WINTER_CAPITAL*(1+CAPITAL_CONSTRAINT) then              { Limit the annual change in capital }
           capital := WINTER_CAPITAL*(1+CAPITAL_CONSTRAINT);
 if capital < WINTER_CAPITAL*(1-CAPITAL_CONSTRAINT) then
           capital := WINTER_CAPITAL*(1-CAPITAL_CONSTRAINT);
 prod_mult := 1 + (1.5 * (summer_rain[Year]-mean_summer_rain)/mean_summer_rain);   { Calculate fresh growth: }
 if prod_mult < 0.05 then prod_mult := 0.05;
 fresh_growth_1 := fresh_growth;
 fresh_growth := prod_mult * INTEREST_RATE * capital;
 food_resource := edibility * capital + fresh_growth;             { Calculate K = food supplies around rocks}
 std_resource := interest_rate * winter_capital + edibility * winter_capital;
 KC := std_KC * food_resource/std_resource;
END;

procedure dassie_reproduce(year:INTEGER;K:REAL);
{
=====================================================================================================
                              HYRAX  LEVEL
        This procedure models the events in the hyrax population. Carrying capacity is used to determine the age-specific
        mortality and fecundity rates.. Age-specific predation is also included here
=====================================================================================================
}
VAR upper_range, lower_range :REAL;
BEGIN
 K_ratio := No_Hyrax_prev/K_prev;                     { Ratio: (current_#_dassies)/(prev. years' carrying capacity) }
 Diff_K := Abs(K_prev-No_Hyrax_prev);
 P_ratio := K_ratio;                                  { remove ratio at this stage}
 if K_ratio > K_factor then K_ratio := K_factor       { compare ratio with level at which max/min mortality rates set in}
 else if K_ratio < (1/K_factor) then K_ratio := 1/K_factor;
 If K_ratio >= 1 then                                 { Scale K_ratio }
   K_ratio := 0 - (K_ratio - 1)/(K_factor - 1)
 else
   K_ratio := (1 - K_ratio) / (1 - (1/K_factor));
 for i:=0 to 12 do begin                              { For each hyrax age class: }
   incmort_F[i] := incmort_F_av[i] - K_ratio * (incmort_F_av[i]-incmort_F_gd[i]);   { calculate age-specific mortalities, }
   incmort_M[i] := incmort_M_av[i] - K_ratio * (incmort_M_av[i]-incmort_M_gd[i]);
   othmort_F[i] := othmort_F_av[i] - K_ratio * (othmort_F_av[i]-othmort_F_gd[i]);
   othmort_M[i] := othmort_M_av[i] - K_ratio * (othmort_M_av[i]-othmort_M_gd[i]);
   Pregnancy[i]  := Preg_av[i] + K_ratio * (Preg_gd[i]-Preg_av[i]);     { age-specific pregnancy rate, }
   LitterSize[i]  := Litr_av[i] + K_ratio * (Litr_gd[i]-Litr_av[i]);    { age-specific litter size, }
   incdie_F[i] := incmort_F[i] * F_prev[i];                            { total number of female hyrax dying }
   othdie_F[i] := othmort_F[i] * F_prev[i];                            { non-eagle mortality }
   incdie_M[i] := incmort_M[i] * M_prev[i];                            { (-M-suffix = MALE; -F = FEMALE) }
   othdie_M[i] := othmort_M[i] * M_prev[i];
 end;
 R_diff := gestn_rain[year]-mean_gestn_rain;                          { Assess gestation rainfall }
 upper_range := max_gestn_rain - mean_gestn_rain;
 lower_range := mean_gestn_rain - min_gestn_rain;
 if R_diff >= 0 then                                                  { calculate foetal resorption in Good year: }
   resorption := FETAL_RESORPTION - (R_diff/upper_range) * (FETAL_RESORPTION - RESORPTION_GOODyr)
 else                                                                 { In Bad year: }
   resorption := FETAL_RESORPTION + (R_diff/lower_range) * (RESORPTION_BADyr - FETAL_RESORPTION);
 foet_mult := gestn_rain[year]/mean_gestn_rain;
 if foet_mult > 1.4 then foet_mult := 1.4;
 if foet_mult < 0.6 then foet_mult := 0.6;
 eagle_predation;                                                     { call in eagle predation here }
 lynx_predation;                                                      {  call in caracal predation here }
 alt_demand := eagle_alt + lynx_alt;                                  { calculate predator demand for alternate prey }
```

```
{ ─────────────────────────────────────────────
      TO DECIDE ON FINAL MORTALITY RATE DEPENDING ON PREDATOR REMOVAL ETC.
   Choose model version        1. inclusive motality
                               2. separate (remove?) eagle mortality from other mortality
                               3. impede predators
  ─────────────────────────────────────────────────────── }
 if version = 1 then
   for i:=0 to 12 do begin
     actmort_M[i] := incmort_M[i];                          { all mortality modelled inclusively }
     actmort_F[i] := incmort_F[i];
   end;
 if version = 2 then
  for i:=0 to 12 do begin
    if remove_eagle_effect = FALSE then begin
      actmort_M[i] := M_eagle_kill[i]/M_prev[i] + othmort_M[i];     { model eagle mortality and other mortality separately }
      actmort_F[i] := F_eagle_kill[i]/F_prev[i] + othmort_F[i];
      end
    else begin                                                       { remove eagles  - captured hyrax do not die }
      actmort_M[i] := othmort_M[i] + Comp_old*M_eagle_kill[i]/M_prev[i];   { except for a small portion dying of old age etc. }
      actmort_F[i] := othmort_F[i] + Comp_old*F_eagle_kill[i]/F_prev[i];
      end
  end;
 if version = 3 then
  for i:=0 to 12 do begin                                            { both predators impeded by X% - }
     actdie_M[i] := ((1-impede) * M_eagle_kill[i]) +  (othdie_M[i] - (impede * M_lynx_kill[i]));     { X% of would be kills do not die }
     actmort_M[i] := actdie_M[i]/M_prev[i];
     actdie_F[i] := ((1-impede) * F_eagle_kill[i]) +  (othdie_F[i] - (impede * F_lynx_kill[i]));
     actmort_F[i] := actdie_F[i]/F_prev[i];
  end;
{=========== ALL MODEL VERSIONS : ===========}

 if P_ratio > crash_level then begin                                 { Heavy compensatory mortality due to starvation / disease .. }
   inc(no_crashes);                                                  { when P far exceeds K }
   for i:= 0 to 12 do begin
         actmort_F[i] :=  crash_mort;
         actmort_M[i] := crash_mort;
         end;
   end;
 no_babies := 0;                                                     { Accumulate populations statistics: }
 No_Females := 0;                                                    { initialise accumulators }
 No_Males := 0;
 No_Hyrax_next := 0;
 total_dying := 0;
 for i:=1 to 12 do begin
     M_next[i] := M_prev[i-1] * (1-actmort_M[i-1]);                  { Implement mortality rates that were decided upon above }
     F_next[i] := F_prev[i-1] * (1-actmort_F[i-1]);
     No_babies := No_babies + F_next[i] * Pregnancy[i] * LitterSize[i] * (1 - resorption);
     No_Females := No_Females + F_next[i];
     No_Males := No_Males + M_next[i];
     No_Hyrax_next := No_Hyrax_Next + F_next[i] + M_next[i];
     total_dying := total_dying + M_prev[i-1]*actmort_M[i-1] + F_prev[i-1]*actmort_F[i-1];
     end;
 total_dying := total_dying + M_prev[12] + F_prev[12];
 other_dying := total_dying - eagle_kill;
 if bias = TRUE then                                                 { If selected, implement rainfall bias in sex ratio of juveniles }
   F_next[0] := (No_babies/2)*foet_mult                              {    using modifiers calculated for foetal resorption earlier     }
 else
   F_next[0] := No_babies/2;
 M_next[0] := No_babies-F_next[0];
 No_Hyrax_next := No_Hyrax_Next + No_babies;

 pop_growth := (No_Hyrax_next-No_Hyrax_Prev)/No_Hyrax_Prev;
 rec_loss := (2.345*No_Females)-No_babies;                           { measure losses through the upper and lower trophic levels }
 juv_loss := (M_prev[0]-M_next[1]) + (F_prev[0]-F_next[1]);
 loss_above := (eagle_kill + lynx_kill + (0.2*juv_loss));           { # hyrax dying from predation  }
 loss_below := rec_loss + (total_dying-loss_above);                  { # hyrax dying from starvation etc. }
 mort_above := loss_above/No_Hyrax_Prev;
 mort_below := loss_below/No_Hyrax_Prev;
 No_Females := No_Females + F_next[0];
 No_Males := No_Males + F_next[0];
 r := ln(No_Hyrax_next/No_Hyrax_prev);
 SR := (No_Females-F_next[0]-F_next[1])/(No_Males-M_next[0]-M_next[1]);     { Calculate sex ratio }
 fjuv := (F_next[0]+M_next[0])/No_Hyrax_next;                        { Calculate  population composition }
 fimm := (F_next[1]+M_next[1])/No_Hyrax_next;
 f2yr := (F_next[2]+M_next[2])/No_Hyrax_next;
 fads := (F_next[3]+M_next[3]+F_next[4]+M_next[4]+F_next[5]+M_next[5]
        +F_next[6]+M_next[6]+F_next[7]+M_next[7])/No_Hyrax_next;
 fold := (F_next[8]+M_next[8]+F_next[9]+M_next[9]+F_next[10]+M_next[10]
        +F_next[11]+M_next[11]+F_next[12]+M_next[12])/No_Hyrax_next;
 for i:=0 to 12 do begin                                             { Age-specific values for next generation become current }
     F_prev[i] := F_next[i];
     M_prev[i] := M_next[i];
     No_Hyrax_prev := No_Hyrax_next;
     end;
 K_prev := K;
END;
```

```
procedure eagle_predation;
{
==========================================================================================
                              CALCULATE LOSSES TO EAGLES:
==========================================================================================
}
BEGIN
No_HP := No_Hyrax_prev;
if No_HP > 2*std_KC then No_HP :=2*std_KC;                       { temporary constraints on the number of hyrax available }
if No_HP < 1200 then No_HP := 1200;
if No_HP > std_KC then eagle_kill := 175 + 175*(No_HP - std_KC)/std_KC    { calculate number captured in direct relation to abundance }
          else eagle_kill := 175 - 105*(std_KC - No_HP)/1200;              {   (at low abundance) }
prey_availability := P_ratio;
if prey_availability > 1.4 then prey_availability := 1.4;       { P:K ratio is used here as an indication of hyrax 'availability' }
if prey_availability < 0.6 then prey_availability := 0.6;
eagle_kill := eagle_kill * prey_availability;                   { Modify number taken according to hyrax availability }
if eagle_kill > 350 then eagle_kill := 350;
if eagle_kill < 70 then eagle_kill := 70;                       { constrain the hyrax kill by eagles within indicated limits}
eagle_alt := 220-eagle_kill;                                    { eagle demand for alternate prey }
if eagle_alt < 0 then eagle_alt := 0;

if age_selection = FALSE then                                   { for test-runs of the model assuming no age-selection of hyrax }
  for i:=0 to 12 do begin
    Pref_M[i] := 1;
    Pref_F[i] := 1;
    end;
Sum_C := 0;
for i:=0 to 12 do begin
   M_tmp[i] := M_prev[i] * Pref_M[i];                           { create 'artificial' availability accoding to eagle preferences }
   F_tmp[i] := F_prev[i] * Pref_F[i];
   sum_C := sum_C + F_tmp[i] + M_tmp[i];
   end;
correction := eagle_kill / sum_C;
 for i:=0 to 12 do begin                                        { calculate age-specific predation by eagles by  }
   M_eagle_kill[i] := M_tmp[i] * correction;                    {  'reducing' the artificial composition to conform to the # taken }
   F_eagle_kill[i] := F_tmp[i] * correction;
   end;
END;

procedure lynx_predation;
{
==========================================================================================
                    CALCULATE LOSSES TO  CARACALS
==========================================================================================
}
VAR rainfall_mult, min_growth, max_growth: REAL;
BEGIN
 Surplus := No_Hyrax_prev - K_prev - (0.75*eagle_kill);         { caracal diet is modelled on this hyrax 'surplus'  }
 if Surplus < 0 then Surplus := 0;                              { but eagles are allowed to remove most of their kill first }
 Lynx_Increase := 1.4;                                          { Set bounds to rate of change in caracal numbers }
 Lynx_Decline  := 0.55;
 Lynx_Diet_Min := 0.02;                                         { Set bounds to caracal functional response (% hyrax in diet) }
 Lynx_Diet_Max := 0.50;
 Lynx_Consumption := 463;                                       {kg/caracal/yr}
 Dassie_Meat := 1.122;                                          {kg meat on average hyrax}

 Min_Growth :=0;                                                { fresh growth is used as an indicator of environmental enrichment ... }
 rainfall_mult := 1 + (1.5 * (max_summer_rain - mean_summer_rain)/mean_summer_rain);   { to model the caracal numeric response }
 Max_Growth := WINTER_CAPITAL*(1+CAPITAL_CONSTRAINT) * INTEREST_RATE * rainfall_mult;
 lynx_mult := Lynx_Decline + (Lynx_Increase-Lynx_Decline) *
    (Fresh_Growth_1-Min_Growth)/(Max_Growth-Min_Growth);
 Lynx_Nos := Lynx_Prev * lynx_mult;                             { derive no. of caracals in eagle territory }
 if Lynx_Nos > 7.44 then Lynx_Nos := 7.44;                      { Limit the no. of caracals }
 if Lynx_Nos < 2.39 then Lynx_Nos := 2.39;

 Lynx_Food := Lynx_Nos * Lynx_Consumption;                      { total food consumption by caracals }

 Surplus_Per_Lynx := Surplus / Lynx_Nos;                        { derive the number of surplus hyrax per caracal }
 if Surplus_Per_Lynx > 210 then Lynx_Diet := Lynx_Diet_Max      {  calculate caracal diet according to this surplus }
 else Lynx_Diet := Lynx_Diet_Min + (Lynx_Diet_Max-Lynx_Diet_Min) * (Surplus_Per_Lynx/210);
 Lynx_alt := (1-Lynx_Diet)*(Lynx_Food/Dassie_Meat);             { caracal demand for alternate prey }
 lynx_kill := (Lynx_Diet * Lynx_Food) / Dassie_Meat;            { No. of hyrax killed by caracals }

Surplus := Surplus + (0.75*eagle_kill);                         { estimation of the cost of hyrax surpluses in terms of sheep grazing }
 sheep := Surplus/SHEEP_EQUIV;
 cost := sheep * COST_PER_SHEEP;

 for i:=0 to 12 do begin
  M_lynx_kill[i] := M_prev[i]/No_hyrax_prev * lynx_kill;        { hyrax kill by caracals is calculated ...  }
  F_lynx_kill[i] := F_prev[i]/No_hyrax_prev * lynx_kill;        { assuming non-selective predation }
  if M_lynx_kill[i] > othdie_M[i] then M_lynx_kill[i] := othdie_M[i];   {   constrain caracal kill to less than non-eagle mortality }
  if F_lynx_kill[i] > othdie_F[i] then F_lynx_kill[i] := othdie_F[i];
  end;

END;
```

```
{ ****************************************************************************************************************
  ***********************************  MAIN  ROUTINE     ********************************************
  ****************************************************************************************************************}
begin
ClrScr;
writeln('DASSIE SIMULATION');
writeln;
write('Location (S/B/D/M). . . . . . . . . . . . . . . . . . ? ');          { Read all the program parameters       }
readln(rain_locality);                                                  {   which are typed in by the operator. }
rain_locality := UpCase(rain_locality);
write('Start Date    . . . . . . . . . . . . . . . . . . . . ? ');
readln(start_date);
write('End Date    . . . . . . . . . . . . . . . . . . . . . ? ');
readln(end_date);
write('Population size at start  . . . . . . . . . . . . . . ? ');
readln(Start_Pop);
if Start_Pop = Std_KC then Start_Pop := 0;
write('Deviation of P from K for max mort. rates etc.  . . . ? ');
readln(K_factor);
write('Deviation of P from K for crash   . . . . . . . . . . ? ');
readln(crash_level);
write('Crash mortality rate    . . . . . . . . . . . . . . . ? ');
readln(crash_mort);
write('Phytomass increment factor in wettest years . . . . . ? ');
readln(max_mult);
write('Phytomass decline factor in driest years  . . . . . . ? ');
readln(min_mult);
write('More male babies in dry years ? (Y/N) . . . . . . . . ? ');
  readln(yn);
  yn := UpCase(yn);
  if yn = 'Y' then bias := TRUE
           else bias := FALSE;
write('Proportion hyrax food needs from less palatables  . . ? ');
readln(unpal_buffer);
writeln('Model version?  1. inclusive mortality');
writeln('                2. separate (remove?) eagle mortality');
writeln('                3. impede predators');
write('                                  . . . . . . . ? ');
readln(version);
if version = 2 then begin
write('Age-selective predation ? (Y/N) . . . . . . . . . . . ? ');
  readln(yn);
  yn := UpCase(yn);
  if yn = 'Y' then age_selection := TRUE
           else age_selection := FALSE;
write('Actually remove eagles ? (Y/N)  . . . . . . . . . . . ? ');
readln(yn);
yn := UpCase(yn);
if yn = 'Y' then remove_eagle_effect := TRUE
         else remove_eagle_effect := FALSE;
if remove_eagle_effect = TRUE then begin
  write('Compensatory mortality for old age etc. . . . . . ? ');
  readln(Comp_Old);
  end;
end;
if version = 3 then begin
write('Predator impedence factor . . . . . . . . . . . . . . ? ');
readln(impede);
end;
writeln;
write('Running ... ');


{ ================================================
          SECTION 1: Read in the demographic parameters:
  ================================================ }

Assign(Demogr_file,'demog.dat');                               { Read demographic data input file: }
reset(Demogr_file);
readln(Demogr_file);                                           { Read file heading with captions }
for i:=0 to 12 do begin                                        { for each age class [i] }
  Readln(Demogr_file, generation,
       M_prev[i],  F_prev[i],                                  { # hyrax to begin with }
       Pref_M[i], Pref_F[i],                                   { eagle preference index }
       incmort_M_bd[i],incmort_M_av[i],incmort_M_gd[i],
       incmort_F_bd[i],incmort_F_av[i],incmort_F_gd[i],
       Preg_bd[i] ,Preg_av[i], Preg_gd[i],                     { pregnacy (bad_year, average_year,good_year) }
       Litr_bd[i], Litr_av[i], Litr_gd[i],                     { Litter size        ditto           }
       othmort_M_bd[i],othmort_M_av[i],othmort_M_gd[i],        { non-eagle mortality rates }
       othmort_F_bd[i],othmort_F_av[i],othmort_F_gd[i]);
  end;
close(Demogr_file);
```

```
if Start_Pop <> 0 then
  for i:= 0 to 12 do begin
    M_prev[i] := Start_Pop * M_Prev[i]/Std_KC;        { Derive hyrax population structure at start }
    F_prev[i] := Start_Pop * F_Prev[i]/Std_KC;
    end;
read_rainfall;                                          { Read the rainfall data from the rainfall file }

assign(outmain_out,'outmain.OUT');                      { Initialise output files and initial values: }
rewrite(outmain_out);
assign(outagest_out,'outagest.OUT');
rewrite(outagest_out);
assign(outagepr_out,'outagepr.OUT');
rewrite(outagepr_out);
capital := 165092;
fresh_growth := winter_capital * interest_rate;
No_Hyrax_prev := std_KC;
No_Hyrax_next := std_KC;
K_prev := std_KC;
Lynx_Prev := Lynx_Avg;

terminate_model := FALSE;
year := start_date - 1;
repeat                                                  { Execute the model year-by-year and write results to the output files: }
  year := year + 1;
  vegetation_grow(year,No_Hyrax_Prev);                  { This determines KC, the carrying capacity }
  if capital < 0  then terminate_model := TRUE;
  write(outmain_out,year:4,summer_rain[year]:6,gestn_rain[year]:6,capital:13:0,fresh_growth:12:0,KC:11:1);

  dassie_reproduce(year,KC);                            { This is where most of the demographic action takes place }
  writeln(outmain_out,No_Hyrax_prev:11:1,r:9:4,total_dying:12:0,eagle_kill:8:1,Lynx_Nos:5:2,Surplus:6:0,
    lynx_kill:6:0,No_babies:5:0,other_dying:8:0,sheep:9:1,cost:10:2,pop_growth:8:4,mort_below:8:4,mort_above:8:4,
    Diff_K:10:1,no_crashes:5);
  write(outagepr_out,year:4,F_eagle_kill[0]:9:1,F_eagle_kill[1]:9:1,F_eagle_kill[2]:9:1,F_eagle_kill[3]:9:1,
    F_eagle_kill[4]:9:1,F_eagle_kill[5]:8:1,F_eagle_kill[6]:9:1,F_eagle_kill[7]:9:1,F_eagle_kill[8]:9:1,
    F_eagle_kill[9]:9:1,F_eagle_kill[10]:9:1,F_eagle_kill[11]:9:1,F_eagle_kill[12]:9:1);
  writeln(outagepr_out,-M_eagle_kill[0]:9:1,-M_eagle_kill[1]:9:1,-M_eagle_kill[2]:9:1,-M_eagle_kill[3]:9:1,
    -M_eagle_kill[4]:9:1,-M_eagle_kill[5]:9:1,-M_eagle_kill[6]:9:1,-M_eagle_kill[7]:9:1,-M_eagle_kill[8]:9:1,
    -M_eagle_kill[9]:9:1,-M_eagle_kill[10]:9:1,-M_eagle_kill[11]:9:1,-M_eagle_kill[12]:9:1);
  write(outagest_out,year:4,No_Hyrax_prev:11:1,F_next[0]:9:1,F_next[1]:9:1,F_next[2]:9:1,F_next[3]:9:1,F_next[4]:9:1,
    F_next[5]:8:1,F_next[6]:9:1,F_next[7]:9:1,F_next[8]:9:1,
    F_next[9]:9:1,F_next[10]:9:1,F_next[11]:9:1,F_next[12]:9:1);
  writeln(outagest_out,-M_next[0]:9:1,-M_next[1]:9:1,-M_next[2]:9:1,-M_next[3]:9:1,-M_next[4]:9:1,
    -M_next[5]:9:1,-M_next[6]:9:1,-M_next[7]:9:1,-M_next[8]:9:1,
    -M_next[9]:9:1,-M_next[10]:9:1,-M_next[11]:9:1,-M_next[12]:9:1,
    SR:7:4,fjuv:7:4,fimm:7:4,f2yr:7:4,fads:7:4,fold:7:4);
 until (year = end_date) or (terminate_model = TRUE);
close(outmain_out);                                     { Close down the program and output files }
close(outagest_out);
close(outagepr_out);
writeln('DONE');
end.
```

APPENDIX 11

POSTAL ADDRESSES OF ALL PERSONS CITED 'PERS. COMM.' OR *'IN LITT.'* IN THIS THESIS

Mr. D.G. Allan, Avian Demography Unit, University of Cape Town, Rondebosch 7700, South Africa.

Mr. M. Anderson, Cape Nature Conservation, Private Bag X6102, Kimberley 8300, South Africa.

Mr. G.E.A. Banfield, Matobo Black Eeagle Survey, 49A, Lawley Road, Suburbs, Bulawayo, Zimbabwe.

Prof. R. Barry Jnr., Biology Dept., Frostberg State University, Frostburg, Maryland 21532-1099, U.S.A.

Mr. S.J. Bekker, Cape Nature Conservation, Priv. Bag X6546, George 6530, South Africa.

Mr. P. Benadie, Karoo National Park, P.O. Box 316, Beaufort West 6970, South Africa.

Dr. P. Benson, Dept. Zoology, Univ. of Witwatersrand, P.O. Wits 2001, South Africa.

Dr. A.F. Boshoff, Cape Nature Conservation, Private Bag 1126, Port Elizabeth 6000, South Africa.

Mr. A. Bowland, 32 Kilburn Ave., Durban, Natal, South Africa.

Dr. C.J. Brown, Directorate of Nature Conservation & Recreation Research, Priv. Bag 13306, Windhoek 9000, Namibia.

Mr. W.S. Clark, 4554 Shetland Green Road, Alexandria VA 22312, U.S.A.

Mr. D. Conroy, 'Mon Desir', Hutchinson, near Victoria West 7080, Cape Province, South Africa.

Mr. R.J.M. Crawford, Sea Fisheries Res. Inst., Roggebaii 8012, Cape Town, South Africa.

Dept. Agric., Middelburg, Private Bag X529, Middelburg 5900, South Africa.

Dr. H. Erasmus, Cape Nature Conservation, Private Bag X6102, Kimberley 8300, South Africa.

Mr. N. Esterhuizen, Cape Nature Conservation, Private Bag X6102, Kimberley 8300, South Africa.

Dr. N. Fairall, Jonkershoek Nature Conservation Station, Private Bag 5D14, Stellenbosch 7600, South Africa.

Mrs. V. Gargett, 12 Charles Veale Drive, West Beach SA 5024, Australia.

Dr. A.C. Kemp, Department of Birds, Transvaal Museum, P.O. Box 413, Pretoria 0001, South Africa.

Dr. R.G. Klein, The University of Chicago, Dept. Anthropology, 1126 East 59th St., Chicago, IL 60637, U.S.A.

Dr. M. Knight, National Parks Board Kimberley Office, P.O. Box 110040, Haddison Park 8306, South Africa.

Dr. J. Lindesay, Climatology Research Group, University of the Witwatersrand, Johannesburg 2001, South Africa.

Dr. R. Liversidge, 92 Central Road, Kimberley 8301, South Africa.

Mr. C. Mocke, Paardefontein, Beaufort West 6970, South Africa.

Mr. L.C. Moolman, Addo National Park, P.O. Box 52, Addo 6105, South Africa.

Mr. A.G. Morgan, P.O. Box 71, Steclestroom 5425, South Africa.

Mr. D. Nel, Kookfontein, Beaufort West 6970, South Africa.

Dr. P. Norton, Cape Nature Conservation, Private Bag 1126, Port Elizabeth 6000, South Africa.

Dr. P. Novellie, National Parks Board Kimberley Office, P.O. Box 110040, Haddison Park 8306, South Africa.

Mr. D.J. Olivier, Dept. Agric., Middelburg, Private Bag X529, Middelburg 5900, South Africa.

Mr. D. Pepler, Dept. Nature Conservation & Forestry, University of Stellenbosch, Stellenbosch 7600, South Africa.

Dr. R.M. Randall, National Parks Board, P.O. Box 774, George 6530, South Africa.

Mr. D. Shearing, Layton, Fraserberg, Cape Province, South Africa.

Prof. W.R. Siegfried, Percy FitzPatrick Institute, University of Cape Town, Rondebosch 7700, South Africa.

Mr. C. Skinner, c/o 44 King Street, Irene 1675, South Africa.

Mr. J.H. Swiegers, Montana, Victoria West 7080, South Africa.

Dr. J.M. Thiollay, Ecole Normale Superieure, CNRS-UA258 Laboratoire d'Ecologie, 46, Rue d'Ulm, 75230 Paris Cedex 05, France.

Ms. F. van Heerden, Wilderness National Park, Private Bag X6528, George 6530, South Africa.

Mr. C. Webber, Geography Dept., Johannesburg College of Education, 27 St. Andrews Road, Parktown 2193, South Africa.

Mr. R. Wilmot, Paardekraal, P.O. Box 1109, Beaufort West, 6970, South Africa.