For office use only	For office use only
T1 ________________	F1 ________________
T2 ________________	F2 ________________
T3 ________________	F3 ________________
T4 ________________	F4 ________________

2012

15th Annual High School Mathematical Contest in Modeling (HiMCM) Summary Sheet
(Please attach a copy of this page to each copy of your Solution Paper.)
Team Control Number: 3652
Problem Chosen: A

Summary

The final goal of our model is to simulate the number of Manitoban elks at the end of a period of time and evaluate if the Manitoban elks can adapt to the environmental conditions in eastern America, more specifically the GSMNP (Great Smokey Mountain National Park). In going about this problem, we constructed a model that predicts the change in the elk population over time.

Our model calculated the change in population by first calculating the birth rate and death rate, then calculating the limit to the elk population in GSMNP: The carrying capacity.

We calculated the birth rate of elks by averaging the elk births in the given data to obtain the arithmetic mean, and calculated the death rate in a slightly more complicated manner of splitting it into two sub-models: An accidental death rate model and a natural death rate model. The former accidental death rate model takes the arithmetic mean of the poaching rate, accident rate, predator attack rate, sickness rate, and an unknown factor rate based on the data given to obtain the probability of an elk dying from environmental factors in any given year. The natural death rate model, on the other hand, simply models the deaths of elks that reach their natural age limit.

We then estimated the carrying capacity of the elk population by rearranging the main function and calculating the arithmetic mean of all data inputted into it, and combined the above models to form our final elk population iterated model.

Next, we constructed a Monte Carlo Simulation Program to test our function for us. The program simulates the elk population's interaction with Nature based on the functions derived in the iterated model for a large number of trials and outputs a statistical average of final population for comparison to our given data. We optimize the program by tweaking input rates and the sickness rate function in order to accurately portray the population growth function.

The original figures were derived by using data from 2001- 2007. By applying the calculated rates to the Monte Carlo simulations, a stunning accuracy of ± 5 elks was achieved. After optimization and modification of the constant rates into functions, the absolute uncertainty of elks was reduced to ± 5 for any set of given statistics (including total poached, total killed by predators, total killed by accidents, total killed by sickness etc.)

The model predicts that the number of elks will increase steadily in the future until it peaks at a limit of 1101, then fluctuate and oscillate slightly around this limit.

Elks into the East

Mathematical Models for the Reintroduction of Elks into Eastern America

2012 HiMCM Problem A
Team #3652

Contents

1. Introduction

Before the arrival of the Europeans in America, North America produced a broad variety of fauna, notably the elks. However, after the colonization movement, the number of traditional animals in North America decreased dramatically. The Eastern Elk became extinct in the early 19th century due to human activity.

Now, scientists are trying to reintroduce elks into eastern North America. There are a number of concerns about the program, but the main concern is if the elk specie itself can adapt to the eastern North American environment. The reintroduced species is the Manitoba Elk. Although the Manitoba Elk is similar to the Eastern Elk, it is smaller in size and suited to the western North America and Canadian environment. Can the Manitoba Elk tolerate typical eastern North American diseases? Can the Manitoba Elk digest eastern North American foods efficiently enough to sustain their growth? Can the Manitoba Elk adapt itself to the eastern North American overall environment?

These questions can be solved through tracking the population change in the Manitoba Elks in the eastern North America region, typically, the GSMNP (Great Smokey Mountain National Park). Through measuring the ability for the Manitoba Elk to survive and reproduce in the GSMNP, it is possible to extrapolate the ability for the Manitoba Elk to survive and reproduce in the eastern North American environment. In addition to the natural change in population, there are also human factors to consider: will the Manitoba Elk in eastern North America be annihilated by human poachers? Will the human-made GSMNP provide significant obstacles for the survival of Manitoba Elks?

Therefore, our model should consider the following factors:

1. The birth rate of the Manitoba Elk
2. The death rate, both natural and accidental, of the Manitoba Elk
3. The limit for the population of Manitoba Elks in GSMNP, or the carrying capacity

Our model will also incorporate a modified virtual environment for elks where we can input the above birth and death rates and the carrying capacity and monitor the progression of the elk population over decades. It must also consider these additional factors:

1. The conversion of the above rates into functioning Java code
2. The accuracy of an object-oriented model
3. The variation in the above rates over time

2. The Population Model

2.1 The Approach

The ultimate goal of this project is to create the model of the elk population within the Great Smokey National Mountain Park for the future using existing data, and we decided to first analyze the factors that determine the births and deaths within the system. However, there are many random factors in this process, such as the rate of sickness, the population of predators, and the rate of poaching etc. presenting gave difficulty to directly model the future of the elk population over time.

It is a well-known fact that a Monte Carlo method, a set of computational algorithms that use repeated random sampling over multiple inputs in a system, is an accurate method of estimation and modeling of problems. We aim to first construct a growth rate model, which is an iterated function, composed of a set of polynomial functions that integrate the various factors determining the birth and death of elks. We then integrate the function into a Java model that plays out the growth and decline of the elk population. Through looping this simulation for tens of thousands of times, the program will return a close estimate of the elk population over a given time period. We can compare our results with the data given in the question: If the output over 2001-2011 is accurate, then we can reasonably infer that the program also calculates the future population accurately.

The end result is a pseudo-Monte Carlo model that is able to simulate the progression of the elk population over, say, two decades. While it isn't completely random in the sense that our iterated function is based on absolute sampling data given in the question, it definitely returns an accurate representation of the elk population given those factors.

2.2 Assumptions

1. *Our model is based on the annual increments of time, with new elks introduced on the first day of the year*. We are given the data of the elk population on an annual basis, and it would be inappropriate to engage in monthly calculations of the growth rate of the population.

2. *All elks have a maximum lifespan of 25 years*. While an elk can die from a series of factors before reaching the maximum lifespan, our model needs to assume a maximum limit on an elk's lifespan to simulate the natural death toll of a population.

3. *A female elk, if pregnant, will always successfully give birth to one baby elk.* The calculation of miscarriage is a minor confounding variable, which would inevitably complicate our model.

4. *Female elks can only become impregnated if they are above or equal to 24 months old.*

This sets a minimum reproduction age according to the nature of the elk. Elks are unable to reproduce before age of sexual maturity.

5. *The natural environment in the Great Smoky National Mountain Park is consistent and stable: there are no natural disasters, including epidemics.* These events are anomalies that could potentially devastate the whole elk population and would cause great fluctuations in this model.

6. *Elk that die during a given year are not factored into the calculation of the birth of new elks that year.* While this may seem obvious, it is important otherwise our calculations will be incorrect. Since our information has a precision of 1 year, it is impossible to know which month did the elk die. This would put an unnecessary complication to our calculation, especially in the calculation of the death rate.

7. *A "mature" elk is an elk that is of or older than 24 months of age*

8. *All elks reintroduced are already mature elks.* This ensures the efficiency of reproduction in the system, and also makes more sense than introducing baby elks into the Great Smokey Mountain National Park, which would be both dangerous and time-consuming.

9. *The male-to-female ratio is very close to 1:1.* This allows us to model the birth rate more easily.

10. *All baby elks are born in autumn.*

11. *The GSMNP does not experience changes in land, resources, climate, and governance.* This makes the carrying capacity stable, and therefore makes the model independent to external factors.

2.3 Variables

Symbols	Explanation
A	the reciprocal of the carrying capacity
N(t)	the population of elks over time (T$\in N^*$) in the year (t+2000)
g(t)	the growth rate in year (t+2000)
R(t)	the number of reintroduced elk in year (t+2000)
B(t)	the birth rate in year (t+2000)
D(t)	death toll in year (t+2000)
r_m	possibility of mating per year per elk
n	number of offspring produced each mate
N'(t)	number of matured elks able to mate in year (t+2000)
AD(t)	death toll due to accident in year (t+2000)
ND(t)	natural death toll in year (t+2000)
r_{ad}	accidental death percentage
N_{base}(t)	the base number for computing the accidental death rate in year (t+2000)

$n_p(t)$	number of elks dead due to poaching activity in year (t+2000)
$n_u(t)$	number of elks dead due to unknown circumstances in year (t+2000)
$n_{ac}(t)$	number of elks dead due to accidents in year (t+2000)
$n_{pr}(t)$	number of elks dead due to predators in year (t+2000)
$n_s(t)$	number of elks dead due to sickness in year (t+2000)
r_{nd}	possibility for an elk to die due to sickness
r_p	possibility for an elk to die due to poaching activity
r_u	possibility for an elk to die due to unknown circumstances
r_{ac}	possibility for an elk to die due to accidents
r_{pr}	possibility for an elk to die due to predators
r_s	possibility for an elk to die due to sickness
R_{al}	Possibility of an elk surviving a time cycle

2.4 The Model

A population grows simply if the number of births each year exceeds the number of deaths in that year. Based on this reasoning, our model defines the growth of the population simply as the arithmetic difference between the birth rate and the death rate within the elk group.

The calculation of the birth and death rate in an elk population necessitates the creation of further sub-models, in which we pinpoint and determine the effect of factors that respectively affect the birth and death rate.

$$g(t) \equiv B(t) - D(t)$$

Equation 1.1

In order to truly simulate natural growth of the population of elks, we must consider the factors that are specific to the elks' adaptations and environment. One important factor is the "carrying capacity" of GSMNP, which is the relative maximum population of elks the land area of GSMNP can support with its limited resources. As the population approaches the carrying capacity, the scarcity of resources imposes a natural limit on the elks and the weaker individuals will die out.

This first function, thus, will be composed of a set of polynomial functions:

$$N(t) = N(t-1) + g(t) \ + R(t) - A * N(t-1)^2$$

Equation 1.2

Where the population in year t (N(t)) is equal to the sum of the population in year (t-1) (N(t-1)) the isolated net growth in year t (g(t)), and the amount of Manitoban elk reintroduced in year t (R(t)). The last factor ($A * N(t-1)^2$) is the adjustment to the population based on the logistics model of carrying capacity.

While it may seem more efficient to express this function in terms of the derivative of population over time, denoted as

$$\frac{dP}{dt} = kP - AP^2$$

Equation 1.3

where P represents the total elk population, we must keep in mind that the iterated function is a polynomial whole number function, where we measure growth in terms of the unit year.

2.4.1 Birth Rate Model

This sub-model establishes the birth rate function of a population of elk, which we can define as a function that estimates the percentage of baby elks, B(t), born in a year as compared to the total population of mature elks from the previous year. Thus we need to first derive the function for the number of mature elks through exclusion.

We have already defined, in the assumptions, that mature elks are elks that are older than 24 months old, and therefore must exclude the number of elks born in the last two years who are still alive by multiplying the factorR_{al}, which represents the factor of those elk born still alive, to it.

$$R_{al} = 1 - (rp + ru + ra + rpr + rs) \qquad \textit{Equation 2.1}$$

$$N'(t) = N(t-1) + R(t) - R_{al} * (B(t-1) + B(t)) \qquad \textit{Equation 2.2}$$

In common terms, the mature population on any given year is the sum of the total population from the previous year, the number of elk introduced this year, and the negative of the surviving elk born within the last two years.

Based on our assumption that the male and female elk population is always close to a constant ratio of 1:1, we simply divide the number of mature elks by two to obtain the number of pairs of elks who attempt to reproduce over the year.

$$B(t) = 0.5 * N'(t) * r_m * n$$

Equation 2.3

The population born in year t is thus equal the product of the pairs of mature elks, the rate of pregnancy, and the number of offspring produced per pair.

According to assumption 3, the n is 1. To compute r_m, plug in all the other numbers to get a function of r_m.

According to given statistics, calculations can derive the r_m in each year according to the equation. By applying the calculated r_m, our model produces an average r_m:

Table 1: Calculation of r_m

t	N(t)	B(t-1)	B(t)	N'(t)	n	$r_m(t)$
1	27	0	4	23	1	34.78%
2	51	4	2	45	1	8.89%
3	57	2	10	46	1	43.48%
4	53	10	8	36	1	44.44%
5	56	8	10	39	1	51.28%
6	67	10	13	46	1	56.52%
7	86	13	19	56	1	67.86%
8	96	19	19	61	1	62.30%
9	107	19	19	72	1	52.78%
10	124	19	25	84	1	59.52%
					r_m	48.19%

r_m in this case is 48.19%.

2.4.2 Death Rate Model

This model establishes the death rate function of a population of elk over time. We observed that elk deaths can be split into two main areas: Accidental deaths as a result of natural elements such as predators, sicknesses and poaching, as well as natural deaths resulting from age (25 years old).

$$D(t) \equiv AD(t) + ND(t)$$

Equation 3.1

The death toll in year t is defined as the accidental death toll in year t plus the number of elks that experienced natural death (age reaches limit).

2.4.2.1Accidental Death Rate Model

The model first established the total accidental death rate r_{ad} as the sum of the individual death rates r_p, r_s, r_{pr}, r_a, and r_u. These individual rates denote the percentage of the elk population killed through poaching, through sicknesses, through predator attacks, through accidental events, and through unknown causes respectively. The last two, however, deserve special attention: Some elks die from random accidents (e.g. falls, starvation), and a very small number of elks die for completely unknown reasons, which could be explained as a sampling error from the Department of Wildlife. The model must still factor this unknown death rate into the function in order to create a reliable output.

We calculated the respective rates of death by computing the statistical mean over the eleven years of data presented to us in the question.

The base number of each rate in year t is the sum of the number of elks alive in year (t-1) plus the number of elks reintroduced in year t plus the number of elks born in year t.

$$\mathrm{Nbase}(t) = \mathrm{N}(t-1) + \mathrm{R}(t) +* \mathrm{B}(t)$$

Equation 4.1

Then the model divides the number of elks dead due to each circumstance by the base number. The average is then calculated:

Rate of poaching:

$$rp = \overline{\frac{np(t)}{Nbase(t)}}$$

Equation 4.1.1

The calculation is presented as followed:

Table 2: Calculation of r_p

t	N(t-1)	R(t)	B(t)	$N_{base}(t)$	$n_p(t)$	$r_p(t)$
1	0	25	4	29	0	0.00%
2	27	27	2	56	0	0.00%
3	51	0	10	61	0	0.00%
4	57	0	8	65	1	1.54%
5	53	0	10	63	0	0.00%
6	56	0	13	69	0	0.00%
7	67	5	19	91	0	0.00%
8	86	0	19	105	0	0.00%
9	96	0	19	115	1	0.87%
10	107	0	25	132	1	0.76%
					r_p	0.32%

The r_p according to the calculations shown above is 0.32%.

The average is used in the calculation of the poaching rate because the act of poaching is random and should remain a constant rate throughout the years if the government doesn't act to decrease the rate of poaching, assuming that the environment is consistent[1].

[1] See assumption 5 for details

Rate of sickness:

$$rs = \overline{\frac{ns(t)}{Nbase(t)}}$$

Equation 4.1.2

The calculation is presented as followed:

Table 3: Calculation of r$_s$

t	N(t-1)	R(t)	B(t)	$N_{base}(t)$	$n_s(t)$	$r_s(t)$
1	0	25	4	29	1	3.45%
2	27	27	2	56	5	8.93%
3	51	0	10	61	2	3.28%
4	57	0	8	65	6	9.23%
5	53	0	10	63	2	3.17%
6	56	0	13	69	0	0.00%
7	67	5	19	91	0	0.00%
8	86	0	19	105	3	2.86%
9	96	0	19	115	2	1.74%
10	107	0	25	132	3	2.27%
					r_s	3.49%

The r_s according to the calculations shown above is 3.49%.

The constant rate of sickness is appropriate if we assume that adaptation to the environment does not occur and the environment itself is does not change. Although living in a constant environment, the elks would not be immune to diseases even if groups of elks are repeatedly infected by the diseases. Under this assumption, it is safe to conclude that the rate of sickness should be modeled by the average because every year's sickness rate is too random for linear regression.

Rate of accidents:

$$rac = \overline{\frac{nac(t)}{Nbase(t)}}$$

Equation 4.1.3

The calculation is presented as followed:

Table 4: Calculation of r$_{ac}$

t	N(t-1)	R(t)	B(t)	$N_{base}(t)$	$n_{ac}(t)$	$r_{ac}(t)$
1	0	25	4	29	0	0.00%
2	27	27	2	56	0	0.00%
3	51	0	10	61	1	1.89%
4	57	0	8	65	0	0.00%
5	53	0	10	63	0	0.00%
6	56	0	13	69	1	1.69%
7	67	5	19	91	1	1.32%
8	86	0	19	105	1	1.11%
9	96	0	19	115	0	0.00%
10	107	0	25	132	2	1.77%
					r_{ac}	0.78%

The r_{ac} according to the calculations shown above is 0.78%.
The rate of accidents is also modeled by the average because accidents are too random in nature for presentation in a linear regression -- Accidents cannot be avoided and controlled.

Rate of predator attacks:

$$rpr = \overline{\frac{npr\,(t)}{Nbase(t)}}$$

Equation 4.1.4

The calculation is presented as followed:

Table 5: Calculation of r_{pr}

t	N(t-1)	R(t)	B(t)	$N_{base}(t)$	$n_{pr}(t)$	$r_{pr}(t)$
1	0	25	4	29	1	3.45%
2	27	27	2	56	0	0.00%
3	51	0	10	61	1	1.64%
4	57	0	8	65	2	3.08%
5	53	0	10	63	5	7.94%
6	56	0	13	69	1	1.45%

7	67	5	19	91	4	4.40%
8	86	0	19	105	0	0.00%
9	96	0	19	115	2	1.74%
10	107	0	25	132	0	0.00%
					r_{pr}	2.37%

The r_{pr} according to the calculations shown above is 2.37%.

The rate of predator activity is modeled by the average as well because predator activities are too random in nature to be presented in a linear regression.

Rate of unknown deaths:

$$ru = \overline{\frac{nu(t)}{Nbase(t)}}$$

Equation 4.1.5

The calculation is presented as followed:

Table 6: Calculation of r_u

t	N(t-1)	R(t)	B(t)	$N_{base}(t)$	$n_u(t)$	$r_u(t)$
1	0	25	4	29	0	0.00%
2	27	27	2	56	0	0.00%
3	51	0	10	61	0	0.00%
4	57	0	8	65	3	4.62%
5	53	0	10	63	0	0.00%
6	56	0	13	69	0	0.00%
7	67	5	19	91	0	0.00%
8	86	0	19	105	5	4.76%
9	96	0	19	115	3	2.61%
10	107	0	25	132	2	1.52%
					r_u	1.35%

The r_u according to the calculations shown above is 1.35%.

The rate of unknown deaths is modeled by the average too because these type of deaths occur in an arbitrary and uncontrollable rate and are not appropriate to be presented in a linear regression.

Total rate of accidental death:

The total rate of death from any accidental factor in nature is the sum of the death rates mentioned above:

$$\mathrm{rad} = \mathrm{rp} + \mathrm{rs} + \mathrm{rac} + \mathrm{rpr} + \mathrm{ru}$$

Equation 4.1.6

Applying the equation presented above, we can derive the accidental death rate as followed:

$$rad = 0.32\% + 3.49\% + 0.78\% + 2.37\% + 1.35\%$$
$$= 8.31\%$$

In any given year, this factor is applied to all elks surviving from the previous year, all elks introduced this year, and one-quarter of the elks born in that year (since they are born in September and are exposed to the environment for only three months, which is a quarter of a year). Therefore the final equation for accidental death toll, based on above data, is:

$$AD(t) = rad * [N(t-1) + R(t) + 0.25 * B(t)] = 0.0831 * [N(t-1) + R(t) + 0.25 * B(t)]$$

2.4.2.2 Natural Death Rate Model

The natural death toll is much simpler: Any surviving elk that reaches 25 years old will pass away, calculated as the rate of an elk surviving to 25 years old multiplied by the number of elks born 25 years ago.

$$\mathrm{ND(t)} = \mathrm{rnd} * \mathrm{B(t-25)}$$

Equation 4.2.1

Therefore the final equation for the death toll function is:

$$D(t) = (rp + ru + ra + rpr + rs) * [N(t-1) + R(t) + 0.25 * B(t)] + \mathrm{rnd} * \mathrm{B(t-25)}$$

Equation 4.1.2

2.4.3 Estimation of Carrying Capacity

The carrying capacity is essential in any calculation or modeling of the nature. It measures the maximum capacity of the land for a specific species in the most favorable conditions in the given environment. The A factor is isolated from the equation above ($N(t) = N(t-1) + g(t) + R(t) - A * N(t-1)^2$), and substituted with the birth rate and death rate equations, resulting in:

$$A = \frac{(1 - r_d) * N(t-1) - \left(\frac{r_m * r_{ad}}{2} + \frac{r_{ad}}{4}\right) * B(t) - \frac{r_{ad}}{4} * B(t-1) + (1 - r_{ad}) * R(t)}{N(t-1)^2}$$
$$= \frac{0.9073N(t-1) - 0.45511065B(t) - 0.0223453B(t-1) + 0.9073R(t)}{N(t-1)^2}$$

Input the values of the numbers for the given years, we have:

Table 7: Calculation of A

t	N(t-1)	N(t)	B(t-1)	B(t)	R(t)	A(t)
1	0	27	0	4	25	n/a
2	27	51	4	2	27	n/a
3	51	57	2	10	0	0.000964
4	57	53	10	8	0	0.003355
5	53	56	8	10	0	0.001761
6	56	67	10	13	0	-0.00028
7	67	86	13	19	5	n/a
8	86	96	19	19	0	0.000523
9	96	107	19	19	0	0.000498
10	107	124	19	25	0	0.000122
11	124	140	25	19	0	0.000313
$\overline{A(t)}$						0.000908
$\frac{1}{\overline{A(t)}}$						1101.811

Some values are presented as N/A because they give results that are bigger than 1 so these values are omitted in the calculation (because A should be bigger than 1 and smaller than 0 to make $\frac{1}{\overline{A(t)}}$ bigger than 1). The formula for calculating the carrying capacity is:

$$carrying\ capacity \equiv (\bar{A})^{-1}$$

Equation 5.1

An average is used for estimation for the carrying capacity. This average gives out a value of 1101.8, which is rounded by decimal omission giving 1101. This carrying capacity serves as a limit for the elk population in our model. However, since this is only estimation, there could be years where the number of elks exceeds the carrying capacity. As the population of elks approach the carrying capacity, the death rate will rise exponentially due to overconsumption of resources causing a dramatic drop in the population, usually decreasing the number of elks to below carrying capacity, in the near future. The decreased population will then once again have a higher growth rate and exceed the carrying capacity once more. This cyclic process occurs rather rapidly with small amplitude.

2.5 Model Testing: The Monte Carlo Simulation

2.5.1 Introduction and Rationale

Once we have constructed the mathematical description for population growth, we apply it by constructing an object-oriented program that creates the object "elk" then instantiate it in a programmed virtual environment. Computer programming is a direct extension of mathematics and is equally confusing: We first create the object class ELK, which simulates the basic properties that apply to a real elk. The program will handle this object internally and create a data container to store instances of this object. Each object will represent one elk[2]. Then we create the class NATURE, which contains the factors of environment such as sicknesses and predators that can act upon the ELK object. Finally, we create the main function HIMCM that runs the course of the Elk society five hundred thousand times, compiles the results, and outputs it onto a spreadsheet file. For the purpose of this model and limitations due to computational power, time passes in increments of months to ensure a more realistic and economic depiction of natural events. Post-analysis on the model includes optimization of code by modifying the sickness rate into a logarithmic function, and making further adjustments to specific rates by comparing the output data with the original data given in the question.

2.5.2 Additional justifications

1. *All original formulas from the iterated function are used.*
2. *The sample data given in the question contains no errors.*
3. *The birth rate and death rates are constant factors.*
4. *Environmental factors only apply when the elk population peaks carrying capacity.*
5. *A Gaussian distribution was added to the "Strength" property of the elks to model the overall strength of an elk through the course of its life.*

2.5.3 Objects and Properties

Our model is based on the construction of objects and the modification of its properties. Here we construct the object ELK, with the properties as follows:

1. **Health(Double)**: A randomized number between 0-50 which can be decreased by various Java methods. Once an elk's health falls beneath 10 HP, it dies.

[2] The instances of elk here can be both dead and alive. The data container essentially stores all the elks that are involved

2. **Strength (Double)**: A randomized number with a base value between 0-50 that varies over time[3] and determines the susceptibility of an elk to natural factors.
3. **Alive (Boolean)**: The dead/alive state of the elk. When it dies, the program stops applying methods to it anymore.
4. **Pregnant (Boolean)**: Female elks have the ability to become pregnant, which will be explained in the *pregnancy* method.
5. **Gender (Boolean)**: Elks are initialized with an equal chance of being male or female using a random number generator, and this determines whether it is able to become pregnant or not.
6. **Age:** Represented in months. Newborn elks start at age 0.

2.5.4 Variables

In addition to the properties of the Elk object, our program also contains many variable inputs that determine the results of the algorithm.

Input Variables

cCap	The carrying capacity of the population
OCDr	The death probability created by overpopulation, calculated as $OCDr = 1 - 1/cCap$
pRate	Pregnancy rate
sRate	Sickness rate
poaRate	Poaching rate
accRate	Accident rate
unkRate	Unknown death rate
predRate	Predator attack rate
trials	The number of trials the user wishes to run

The variable OCDr is the application of the environmental factor.

2.5.5 Methods

The following methods are the functions that act upon the Elk object and modify its properties. For simplicity we have omitted some of the internal methods that serve programming purposes.

Class Elk:

1. **Damage_Health**
   Reduces the health of an Elk.
2. **Damage_Strength**

[3] The Gaussian distribution of strength

Reduces the strength of an Elk.

3. **Age_Strength**
   Returns the increase/decrease in strength of an elk where
   $Increment = 50 * 2.718^{(-0.5*(ag/12-25/2)*(ag/12-25/2))/32)}$
   *ag represents the age of the elk in months.
4. **Older**
   Ages the elk. If the elk reaches 300 months of age, it dies a natural death.
5. **Pregnancy**
   If the elk is female and of age, it has a possibility (determined by an RNG) to become pregnant. When pregnant, a counter keeps count of time and triggers the method Birth at eight months.
6. **Birth**
   Gives birth to a new baby elk.

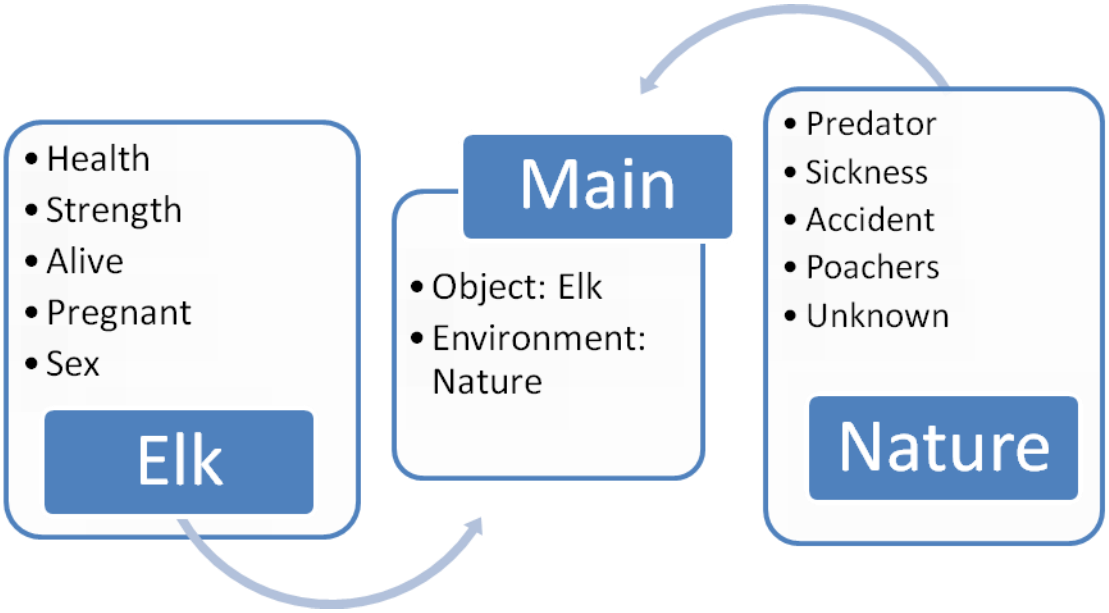
Class Nature:

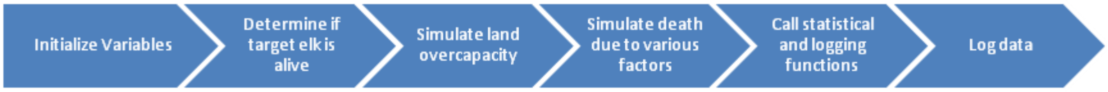
1. **Sickness**
   Compares the strength of a randomly generated sickness to the strength of an elk, and if stronger, damages the elk through methods damage_Health() and damage_Strength().
2. **Oops (The name "oops" is for comedic purposes only)**
   Kills an elk. Combined with the method rndet to simulate the effect of poaching, accidental and unknown deaths.
3. **Predator**
   Compares the strength of a randomly generated predator to the strength of an elk, and if stronger, kills the elk.
4. **Capdmg**
   Simulates the effect of overpopulation through damaging elks randomly.

Class HIMCM:

1. **Rndet(no input needed) – returns a random boolean**
   Runs a random-number generator against a preset rate to determine whether or not an event is triggered. For example, I could run Rndet(0.3) to simulate an event with a 30% chance of occurring.
2. **Lifecheck(the array of elks) – returns to nAlive**
   Checks the "alive" status of each elk in the array and returns the number of elks alive, a.k.a. the size of the population
3. **Introduce(number of elks) – no return**
   Introduces new elks of a randomly generated age between 5 and 20 months old.

Main Function




Initialize Variables
Determine if target elk is alive
Simulate land overcapacity
Simulate death due to various factors
Call statistical and logging functions
Log data

The main function executes the actual program using the series of methods listed above. Over the course of 20 years, with 12 months to the year, the program runs the function *rndet* over each method over each elk in the array. The methods called are linked to the birth and death factor of the iterated function, namely the pregnancy, sickness, predator attack, poaching, accident and unknown rates. It then outputs the individual deaths and the final population of this “trial” to a spreadsheet, and then resets the virtual society for the next “trial”. After a very large number of “trials” (as inputted by the user), the statistical mean for each population value is also computed onto the spreadsheet and expressed as a graph.

2.6 Sensitivity of the Model

To test the sensitivity of the Model, we used a basic method-OFAT, as our model is governed by both polynomial functions and random variables. In a sample run of 20 years, the base numbers are given as followed:

Base Reference

```
Year: 2021
Statistical Averages:
Trials ran: 1000
Population Alive: 484
Sickness: 180
Accidents: 31
Unknown: 97
Poached: 13
Predator Kill: 107
Aged Death: 0
Poach Rate: 0.030833333333333338%
Sickness Rate: 0.40916666666666666%
Accident Rate: 0.06916666666666667%
Pregnant Rate: 7.5%
Unknown Death Rate: 0.21750000000000003%
Predator Kill Rate: 0.23916666666666664%
Carrying Capacity: 1101
```

2.6.1 Sensitivity of the Poached Rate

We increased the poached rate by 1%, and ran the program again. It gave the results as followed:

Increase Poach rate by 1% / year

```
Year: 2021
Statistical Averages:
Trials ran: 1000
Population Alive: 479
Sickness: 180
Accidents: 30
Unknown: 97
Poached: 52
Predator Kill: 107
Aged Death: 0
Poach Rate: 0.11416666666666667%
Sickness Rate: 0.40916666666666666%
Accident Rate: 0.06916666666666667%
Pregnant Rate: 7.5%
Unknown Death Rate: 0.21750000000000003%
Predator Kill Rate: 0.23916666666666664%
```

```
Carrying Capacity: 1101
```

The number of poached elks increased by 300% in the test run , which was close to the percentage increase in the rate(270%), while the final population only had a minimal decrease of 0.4%. This show that the elk population is not very sensitive to the number of poached elks, which is accurate in our model as a change in one of the death rate functions should not cause a very big fluctuation of population as the herd would counteract this increased deaths by having more births.

2.6.2 Sensitivity of the Predator Rate

We increased the predator rate by 1%, and ran the program again. It gave the results as followed:

Increase predator kill rate by 1%/year

```
Year: 2021
Statistical Averages:
Trials ran: 1000
Population Alive: 483
Sickness: 181
Accidents: 31
Unknown: 98
Poached: 14
Predator Kill: 144
Aged Death: 0
Poach Rate: 0.030833333333333338%
Sickness Rate: 0.4091666666666666%
Accident Rate: 0.06916666666666667%
Pregnant Rate: 7.5%
Unknown Death Rate: 0.21750000000000003%
Predator Kill Rate: 0.3225%
Carrying Capacity: 1101
```

The number of predator killed elks increased by 34.6%, almost equivalent to percentage increase in the rate(34.8%). The final population didn’t decrease but actually increased by 0.6%. This shows that the elks are counteracting the predators in giving more births, and is consistent with nature. This also shows that the population alive is not sensitive towards the increase of the death rate, which is true because of the counteraction.

2.6.3 Sensitivity of the Accident Rate

Increase Accident rate by 1%/year

```
Year: 2021
Statistical Averages:
Trials ran: 1000
Population Alive: 477
Sickness: 178
Accidents: 68
```

```
Unknown: 96
Poached: 13
Predator Kill: 106
Aged Death: 0
Poach Rate: 0.030833333333333338%
Sickness Rate: 0.40916666666666666%
Accident Rate: 0.1525%
Pregnant Rate: 7.5%
Unknown Death Rate: 0.21750000000000003%
Predator Kill Rate: 0.23916666666666664%
Carrying Capacity: 1101
```

The number of elks killed by accidents increased about 127%, close to the percentage increase in the rate(120%). The final population decreased about 0.6% but counteraction was definitely seen through the process, which is consistent with both our model and nature. This also shows that the population alive is not sensitive towards the increase of the accident rate, which is true because of the counteraction.

2.6.4 Sensitivity of the Sickness Rate

Increase Sickness Rate by 1%/year

```
Year: 2021
Statistical Averages:
Trials ran: 1000
Population Alive: 481
Sickness: 217
Accidents: 31
Unknown: 97
Poached: 13
Predator Kill: 106
Aged Death: 0
Poach Rate: 0.030833333333333338%
Sickness Rate: 0.4925%
Accident Rate: 0.06916666666666667%
Pregnant Rate: 7.5%
Unknown Death Rate: 0.21750000000000003%
Predator Kill Rate: 0.23916666666666664%
Carrying Capacity: 1101
```

The number of elks killed by sickness increased about 20.6%, very close to the percentage increase in the rate (20.4%). The final population increased about 0.2%, which shows that counteraction does appear in the model, and it is consistent with our model and nature. This also shows that the population alive is not sensitive towards the increase of the sickness rate, which is true because of the counteraction.

2.6.5 Sensitivity of the Unknown Rate:

Increase unknown deaths rate by 1%/ year

```
Year: 2021
Statistical Averages:
Trials ran: 1000
Population Alive: 476
Sickness: 179
Accidents: 30
Unknown: 134
Poached: 13
Predator Kill: 106
Aged Death: 0
Poach Rate: 0.030833333333333338%
Sickness Rate: 0.4091666666666666%
Accident Rate: 0.06916666666666667%
Pregnant Rate: 7.5%
Unknown Death Rate: 0.30083333333333334%
Predator Kill Rate: 0.23916666666666664%
Carrying Capacity: 1101
```

The number of elks killed by sickness increased about 38.1%, very close to the percentage increase in the rate (38.3%). The final population decreased about 0.8%, which shows that counteraction does appear in the model, and it is consistent with our model and nature. This also shows that the population alive is not sensitive towards the increase of the unknown death rate, which is true because of the counteraction.

2.6.6 Sensitivity of the Pregnancy Rate:

Increase Pregnant Rate by 1% / 6 months

```
Year: 2021
Statistical Averages:
Trials ran: 1000
Population Alive: 498
Sickness: 184
Accidents: 31
Unknown: 99
Poached: 14
Predator Kill: 109
Aged Death: 0
Poach Rate: 0.030833333333333338%
Sickness Rate: 0.4091666666666666%
Accident Rate: 0.06916666666666667%
Pregnant Rate: 7.666666666666668%
```

Unknown Death Rate: 0.21750000000000003%
Predator Kill Rate: 0.23916666666666664%
Carrying Capacity: 1101

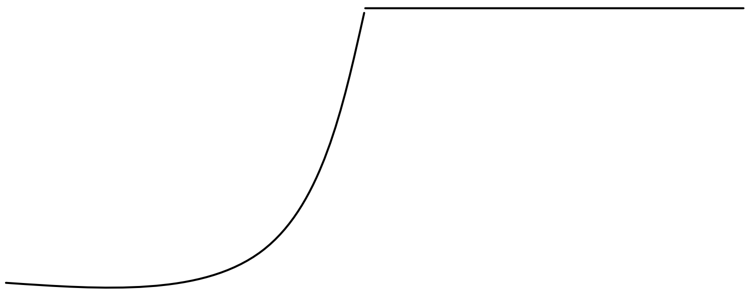
Although the pregnant rate per month only increased by 2.1%, the final population growth increased by 3.8%, which was greater than the growth in the pregnancy rate. This shows that our model is sensitive towards the pregnancy rate, which is consistent with our model as the pregnancy rate is very important in determining the growth rate of the elks and therefore deciding the number of the elks.

2.7 Strengths and Weaknesses

According to our results, the population model gives a thorough and precise estimation of the population. Through the comprehensive analysis of the growth rate, the model splits the model into several sub-models that form up the total model. Therefore, the model is accurate, precise, and extensive.

The computer model has considered every aspect included in the growth rate model mentioned earlier, and also included the concept of growing limit, which states that the elks cannot increase in numbers infinitely, to make the model more realistic and practical. Computational results mostly coincide with the given statistical data.

However, the model has not considered the process of adaption to the environment. As time passes and the number of elks increases, sickness rate and predator kills will both drop because the elks will adapt to the environment as time passes, thus causing the elks to become more immune to diseases and capable of escaping from predators. The employment of the Monte Carlo simulation inevitably generates copious amounts of data that may become inefficient due to the required processing power and data analysis required. Last but not least, the model did not fully apply the concept of carrying capacity into action, which makes the graph of the population of elks reach a plateau with sharp inflection near the top look like:



This type of function is not very realistic as the population of a species usually follows a general curve that is similar to the Logistic model, with a clear inflection point when the population is approximately half of the carrying capacity. Our model, however, shows that the inflection point is very near the carrying capacity.

3. Model Improvements

3.1 The New Environmental Factor Model

This model is designed to calculate environmental impact on the elk population, more specifically the reduction in population growth as the population approaches maximum carrying capacity of the Great Smokey Mountain National Park.

As previously stated in the beginning of this model, we attempt to model population growth by using the function

$$N(t) = N(t-1) + g(t) + R(t) - A * N(t-1)^2$$

As the elk population grows, fewer resources are available per elk within the Great Smokey Mountain National Park. Consequently, the elks will face more and more competition, and the growth rate will decrease. We predict that the elk population will peak at and oscillate below a maximum value in this state, namely 1/A (the reciprocal of carrying capacity).

During the test run of the Monte Carlo Model, we found it necessary to introduce another factor,θ, into our environmental factor model, as A was affecting the population too much at the start of the curve. Therefore, our Model turns into:

$$N(t) = N(t-1) + g(t) + R(t) - \theta * A * N(t-1)^{2},$$

The θ is installed because the environmental factor does not immediately affect the population. A coefficient of delay is added to correct for the lag.

Through repeated test runs, the value of θ was settled as 0.16, which in our Java program, is reflected as:

natDead = nAlive * nAlive*0.16*(1/1101); //natDead is the number of deaths due to the environmental factor

where nAlive= $N(t-1)$, $A = \frac{1}{1101}$,and $\theta = 0.16$

3.2 Improvement for Deaths Caused by Sickness

It is stated in the earlier parts of this report that as time passes, the elks become more adapted to the environment and develop immunity for local diseases. However, not all the elks can be immune to a disease completely, which means that there will always be elks that will become sick, which is true according to natural selection. Correcting for the influence of adaptation, the sickness model can be improved by replacing the mean of sickness r_s as a function.

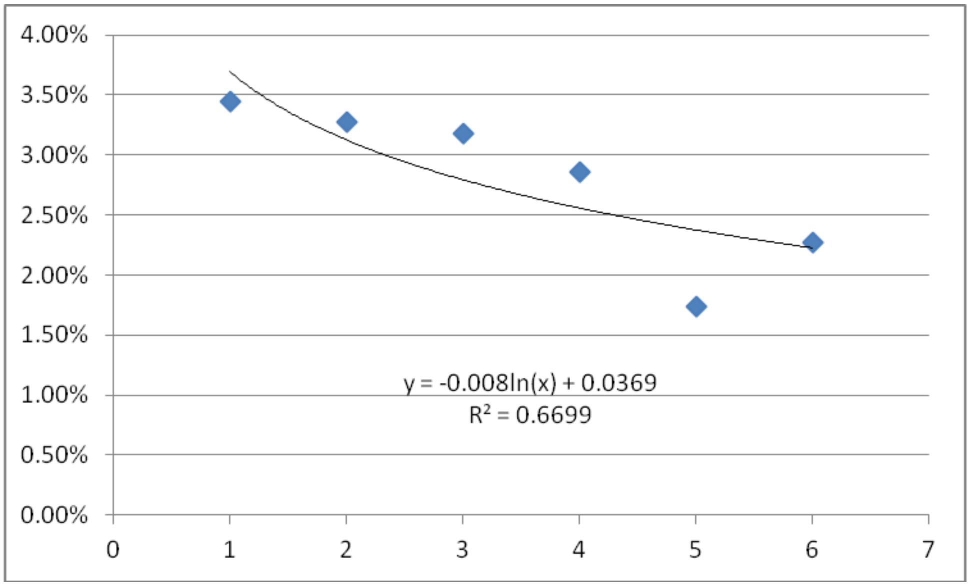
According to the Table 3, there are some outliers in the rate of sickness, which are discarded to improve the overall accuracy of the model. Therefore, the final group of data the model uses is shown as followed:

Table 8: List of acceptable r_s values

t	N(t-1)	R(t)	B(t)	$N_{base}(t)$	$n_s(t)$	$r_s(t)$
1	0	25	4	29	1	3.45%
3	51	0	10	61	2	3.28%
5	53	0	10	63	2	3.17%
8	86	0	19	105	3	2.86%
9	96	0	19	115	2	1.74%
10	107	0	25	132	3	2.27%

After regression, a logarithmic function is introduced to represent the function between r_m and t:

Graph 1: Regression of r_s



The model of $r_m(t)$ would be, according to the graph:

$$rm(t) = -0.008\ln(t) + 0.0369$$

Equation 6.1

This is the improved model for deaths caused by sickness.

3.3 Improvement for Deaths Caused by Predator Activity

Adaptation to elude predators should also be considered. As time passes, the elks will be less susceptible to the threat of predators. However, not all the elks can be strong enough to escape, and there will always be elks that are killed by predators, which is true according to natural selection. Therefore, the predator model can be improved by replacing the constant of predator activity r_{pr} as a function.

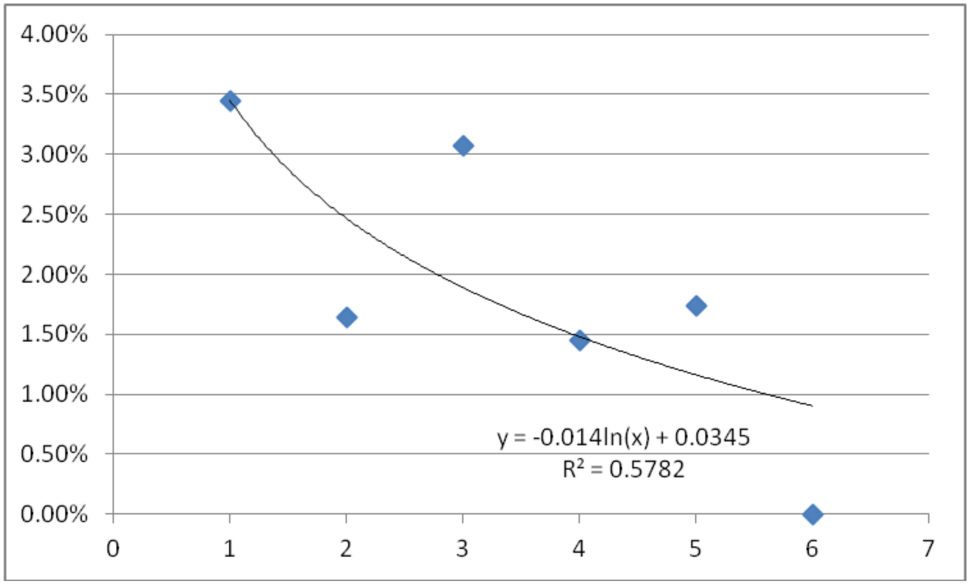
According to the Table 5, there are some outliers in the rate of predator activity That must be dropped to improve the accuracy of the model. Therefore, the final group of data the model uses is shown as followed:

Table 9: List of acceptable r_{pr} values

t	N(t-1)	R(t)	B(t)	$N_{base}(t)$	$n_{pr}(t)$	$r_{pr}(t)$
1	0	25	4	29	1	3.45%
3	51	0	10	61	1	1.64%
4	57	0	8	65	2	3.08%
6	56	0	13	69	1	1.45%
9	96	0	19	115	2	1.74%
10	107	0	25	132	0	0.00%

After regression, a logarithmic function is introduced to represent the function between r_m and t:

Graph 2: Regression of r_{pr}



The model of $r_m(t)$ would be, according to the graph:

$$\mathrm{rm}(\mathrm{t}) = -0.014\ln(\mathrm{t}) + 0.0345$$

Equation 6.2

This is the improved model for deaths caused by predator activities.

4. Growing Plans

To increase the number of elks, there is a need to review the growth function:

$g(t) \equiv B(t) - D(t)$

Therefore, to increase the number, there should either be an increase in the growth rate or a decrease in the death rate. The growth rate, as mentioned, is defined as follow:

$B(t) = 0.5 * N'(t) * r_m * n$

However, the problem is, to increase the growth rate there should be either an increase in the rate of mate or the number of offspring produced each mate. These factors cannot be decided upon human interference, because they are depended on the nature of the elk, which is unchangeable.

Thus we can only try to change the death rate. According to the death rate function:

$$D(t) \equiv AD(t) + ND(t)$$

There can only be a change in the accidental death rate because the natural death rate is the number of the elks still alive that were born in year (t-25). In other words, the natural death rate is dependent on accidental death rate.

To lower the accidental death rate, we need to decrease at least one of the following components of the accidental death rate: r_p, r_{ac}, r_{pr}, r_u, or r_s. According to previous descriptions, the rate of poaching, accidents, and unknown is arbitrary and cannot be avoided, thus there is no possibility to change these death rates. Therefore, we can only influence the death rate due to sickness and the death rate due to predator activity.

4.1 Decreasing r_s

To decrease the sickness rate, the park can inject antibodies to the elks regularly to decrease the possibility of elks being sick. The main threatening disease for elks is the CWD (Chronic wasting disease). Unfortunately, the cause and cure of this disease is still unknown. So there is apparently no way to decrease the death rate by decreasing the sickness rate, and therefore increasing the number of elks. The only way to prevent this disease is to invent a type of medicine in the future days, and then apply it on to the elks.

4.2 Decreasing r_{pr}

To decrease the predator rate, there can be two plans:

1. To lower the number of predators
2. To increase the suitability of the elks towards their predators

However, according to plan 1, once the number of predators decreases, the elk population increases faster, and according to natural selection, the elks die faster too because of more intensive competition. Moreover, the less number of predators in GSMNP would cause a catastrophic influence to the food chain and the environment, so plan 1 is infeasible.

According to plan 2, the main way for increasing the suitability of elks is to increase their strength since the main way of elks surviving from predator attacks is to be fast enough to escape from predator attacks, and the main factor for speed is the strength. To increase the strength of elks, there can be two plans too:

1. To wait for the process of natural selection
2. To inject the elks typical drugs that will increase their strength regularly

For plan 1, the process is not human-controlled, so this plan should be neglected. For plan 2, this is an applicable plan, and can be considered further discussion.

5．Letter to Commissioner

Dear Commissioner,

We are high-school students who are sincerely concerned with the preservation of elks and are actively taking note of the progress made in the Great Smokey Mountain National Park. In turn, we have constructed an iterated function and Java program that simulates the future of the elk population, and are glad to announce that we believe that the elk population will continue to grow in the future.

The first component of our model is the iterated function, in which we have taken the factors determining the birth and death rates of elks inside GSMNP and compiled them into one large function. Using the data given, we calculate that the birth rate is consistently higher than the death rate in the population, and thus the elk population will continue expanding over the next decade or so.

The second component is a Java program that constructs a virtual elk society, and based on inputted environmental factors, simulates multiple futures of the population to return an accurate estimate. The beauty of this program is that you can adjust the program annually to include new data, so the model will become more and more accurate over time.

As with any ecological system, there is a natural limit to the population of a species: The carrying capacity. There will be too little resources per elk to survive, and the large population will also make easy prey for the coyotes and wolves of GSMNP. We have calculated this capacity to be 1101, which means that as the elk population approaches this limit, the growth rate will slowly diminish until it peaks at the carrying capacity. Beyond that point the population will oscillate below the capacity, as nature plays out the natural uncertainties of births and deaths in the population.

Of course, it would be best if we could push the carrying capacity higher so that more elks can populate GSMNP. This can be accomplished through drug injections that make elks stronger and less susceptible to predator attacks, or through developing vaccines to cure the fatal CWD disease.

Commissioner, we greatly appreciate your effort in introducing the Manitoban Elk to the Eastern United States. Despite the differences in disease and environmental tolerance from its larger counterparts, the Eastern elk, the Manitoban Elk seems to be thriving well in the Great Smokey Mountain National Park. We hope that you can take into consideration the implications and ideas presented into our model to continue preserving the wildlife that is essential to nature, the unique world he inhabits.

Yours faithfully,
Team No. 3652

6. Appendix

6.1 Reference

1. *Interesting Facts about Population Growth Mathematical Models*. N.p., 13 Nov. 2000. Web. <http://ejad.best.vwh.net/java/population/facts_math.html>.
2. Br¨, Ake. "Modelling Animal Populations." *Modelling Animal Populations - DiVA - Simple Search*. N.p., 2004. Web. Path: Google; animal population model.

3. Shi, Junping. "Math 410(510) Notes (1) Modeling Population Growth Part 1: single species." *Math 410(510) Notes (1) Modeling Population Growth . - CiteSeer*. N.p., n.d. Web. Path: Google; Math 410(510) Notes (1).
4. *Elk - Wikipedia, the free encyclopedia*. N.p., n.d. Web. <http://en.wikipedia.org/wiki/Elk>.
5. *elk*. N.p., n.d. Web. 18 Nov. 2012. <http://www.handsontheland.org/classroom/04/deer.html>.
6. *Chronic wasting disease - Wikipedia, the free encyclopedia*. N.p., n.d. Web. <http://en.wikipedia.org/wiki/Chronic_wasting_disease>.

6.2 Java Program Code

a. Model 1: Model with constant sickness and predator rates

```
package himcm;
/* Class:Elk
 * Objects: Elk
 * Notes: This class contains the object elk. the elk has properties that can be changed and
manipulated by Nature that will eventually determine whether it survives or not
 */
public class Elk{
//Initialize variables:
   public double bstrength;                         //Initialized strength of elk
   double health;                                   //Health of the elk
   double strength;                                 //Current strength of elk
   int age;                                     //Age of elk
   int PC = 0;                                    //Pregnancy counter (8 months = give birth)
   boolean alive;                                  //State of elk: alive or dead
   boolean gender;                                  //True = male, False = female
   boolean pregnant;                                 //State of pregnancy
 //------------------------------------------------------------------------------------------------------
   public Elk(double Hp, double Str, double Sex) {    //Initialize and define the properties of an
elk object
      health = 50.00*Hp;
      bstrength = 50.00*Str;
      strength = bstrength;
      age = 0;
      alive = true;
      pregnant = false;
      if (Sex<=0.5){
         gender = true;
      }
      else{
            gender = false;
        }
    }

    public Elk() {                                  //Define the default state of and elk
        this(Math.random(),Math.random(),Math.random());
    }

    public void damage_Health(double val){                  //Reduce health of the Elk
        if(val >= 0){
            this.health = this.health - val;
            if(health < 10.00){
                alive = false;
            }
        }
    }

    public void damage_Strength(double val){            //Reduce strength of the elk
        if(val >= 0){
            this.strength = this.strength - val;
        }
    }

    public boolean state(){                         //Return state of the elk object
        return alive;
    }

    public int dispAge(){                           //Return age of the elk object
        return age/12;
    }

    void age_Strength(int ag){                      //Bell curve function of strength in relation to age
        double addent = 50 *(Math.pow(2.718,(-0.5*(ag/12-25/2)*(ag/12-25/2))/32)); //change the
32 to increase the strength of the population
        this.strength = addent +this.bstrength;
    }

    public void older(){                            //Aging process of the elk object
        age++;
        if(age == 300){
            this.alive = false;
            HIMCM.agedeath++;
        }
        this.age_Strength(this.age);
    }

@Override
    public String toString(){                          //Convert object to string output
            return "health: " + health + ", Strength: " + strength + ", gender: " + gender + ", age: " +
age + ", Alive? " + alive + " Pregnancy: "+ pregnant;
    }

    public void pregnancy(Elk[] c, double rand){       //Pregnancy trigger and timer to delay for 8
months
        if(this.alive){
            if((!this.gender)&&(this.age >= 24)&&(!pregnant)&&(HIMCM.month >=1 )){
                if(rand < HIMCM.pRate){    // make sure to change the first (random number) and the
second (pregnancy rate
                    this.pregnant = true;
                }
            }
            if(pregnant){
                PC++;
            }
            if(PC == 12){
                this.birth(c,HIMCM.index);
                PC = 0;
            }
        }
    }

    public void birth(Elk[] e, int SIZE){                       //Function to directly generate one new elk
and initialize it
        pregnant = false;
        e[SIZE] = new Elk();
        HIMCM.index++;
    }
}
```

```
package himcm;
/* Class: Nature
 * Methods: sickness(),oops(),predator(),capdamage()
 * Written by Frank Gu and Andrew Tan
 * Notes: this class includes functions to simulate the natural environmental impacts on the
 *       investigated elk population
 */
public class nature {
    public static void sickness(double dmg, Elk e) {           //Simulates the effect of sickenss
        if (dmg>e.strength){
            e.damage_Health(dmg);
            e.damage_Strength(dmg);
        }
        if (dmg==e.strength){
            e.damage_Health(dmg/2.0);
            e.damage_Strength(dmg/2.0);
        }
    }

    public static void oops(Elk e){                            //Direct public call to kill the elk object
         e.alive = false;
    }

    public static void predator(double stg, Elk e) {           //Simulate predator influence on elk
population by competition
        if (stg > e.strength){
            e.alive = false;
        }
        if (stg == e.strength){
            e.damage_Health(stg);
        }
    }

    public static void capdmg(int pop, int cap, Elk e) {
        if (pop > cap){
            int dmg = (pop - cap)*(pop-cap);
            e.damage_Health(dmg);
            e.damage_Strength(dmg);
        }
    }
```

```
}
/*
 * Standard Model 1A
 * Simulation trials: 1 Million
 * Simulation years: 2001-2026
 */
package himcm;
/**
 * @author Frank Gu and Andrew Tan
 */
import java.io.*;

public class HIMCM {
    static int nAlive = 0;                //The amount of elks currently alive in the population
    static Elk[] potato = new Elk[65535];  //Create the set of Elk objects
    static int index = 0;                 //The index of the elks
    static int cCap = 1101;               //Carrying Capacity
    static double OCDr = 1-1/1101;        //Death probability due to over population
    static double pRate = 0.0396;         //Pregnancy rate
    static double sRate = 0.0491/12;      //Sick rate (Constant for Model 1A)
    static double poaRate =0.0037/12;     //Poaching rate
    static double accRate = 0.0083/12;    //Accident rate
    static double unkRate = 0.0161/12;    //Unknown death rate
    static double predRate = 0.0287/12;   //Predator attack rate
    static int month = 0;                 //Month counter
    static int agedeath = 0;              //Amount of deaths due to old age
    static int k = 1;                     //Number of times to run the simulation
    static int year = 0;                  //Year counter
    static int trials = 100;

public static boolean rndet(double limit) {       //A random function that throws back a boolean
   boolean rn = true;
   double rng = Math.random();
   if (rng>limit){
       rn = false;
   }
   return rn;
}

public static void logData(String path,double pop){  //Prints data onto file(path, population)
```

```
   try{
      PrintWriter out = new PrintWriter(new BufferedWriter(new FileWriter(path, true)));
      out.println(year + "," + pop);
      out.close();
   }
   catch(IOException e){
      e.printStackTrace();
   }
}

public static void logtData(String path,double pop,int s,int a, int u, int p, int pk){  //Prints data
onto file(path, population, sickness,accident,unkonwn,poached,predator kills) by each trial
   try{
      PrintWriter out = new PrintWriter(new BufferedWriter(new FileWriter(path, true)));
      out.println(k + "," + pop + ","+ s + ","+ a + ","+ u + ","+ p + ","+ pk);
      out.close();
   }
   catch(IOException e){
      e.printStackTrace();
   }
}

public static void logln(String path){  //Prints data onto file(path, population,
sickness,accident,unkonwn,poached,predator,da) by each trial
   try{
      PrintWriter out = new PrintWriter(new BufferedWriter(new FileWriter(path, true)));
      out.println("");
      out.close();
   }
   catch(IOException e){
      e.printStackTrace();
   }
}

public static void lifecheck(Elk[] ek) {                    //Check the amount of Elks alive
   nAlive = 0;
   for (int i=0;i<index;i++){
      if (ek[i].alive = true){
         nAlive++;
      }
```

```
    }
}

public static void introduce(Elk[] e,int amount){
    for(int amt = 0; amt < amount; amt++){
        e[index] = new Elk();
        index++;
    }
}

public static void oData(double pop, int s, int a,int u, int p, int pk,int da){            //Programming
debugging output
    System.out.println("Statistical Averages: ");
    System.out.println("Trials ran: " + (k-1));
    System.out.println("Population Alive: " + (int)(pop/k-3));                          //Minus 3 to
calibrate for population alive error
    System.out.println("Sickness: " + (int)(s/k-4));                                    //Minus 4 to calibrate
for sickness error
    System.out.println("Accidents: " + (int)a/k);
    System.out.println("Unknown: " + (int)u/k);
    System.out.println("Poached: " + (int)p/k);
    System.out.println("Predator Kill: " + (int)pk/k);
    System.out.println("Aged Death: " + (int)da/k);
}

public static void main(String[] args) {
    int sick = 0;
    int accident = 0;
    int unknown = 0;
    int poached = 0;
    int predatorkilled = 0;
    double sum = 0.0;
//-----------------------------------------------------------------------------------------------------
    Elk[] potato = new Elk[65535];            //Create a new container for Elks
    introduce(potato,25);                     //Introduce 25 elks to the container
//-------------------------------------------------------------------------------------------------------
    for(;k <= trials; k++){                             //Determine the number of times the cycle is run
        int cnta= 0;              //Sickness Death counter
        int cntb = 0;             //Poach Death counter
        int cntc  = 0;            //Accidental Death counter
    int cntd = 0;               //Unknown cause death counter
    int cnte = 0;               //Predator kill counter

    for(year = 2001; year <= 2026; year++){             //The number of years the sample endures
      if(year == 2002){                          //Introduce 27 elks in the second year
        introduce(potato,27);
      }
      if(year == 2007){                          //Introduce 5 elks in the 7th year
        introduce(potato,5);
      }

      for(month = 1; month <= 12; month++){

        boolean overCap = false;                        //Initialize overcapacity indicator
        lifecheck(potato);                          //Determine nAlive for set potato
        if(nAlive > cCap){                           //Determine whether nAlive is over the capcity
of the land
          overCap = true;
        }

        for(int iElk = 0; iElk < index; iElk++){           //The monthly cycle of one individual
elk begins here
          if((potato[iElk].alive)){                    //First determine if the elk is alive or not
            if((overCap) && (rndet(OCDr))){               //Death with probability due to over
capcity
              potato[iElk].alive = false;
            }

            if((rndet(sRate))&&(potato[iElk].alive)){                           //Death due to
sickness
              potato[iElk].alive = false;
              if(!potato[iElk].alive){
                cnta++;   //Increment death due to sickness counter
              }
            }

            if((rndet(poaRate))&&(potato[iElk].alive)){                          //Death due to
poaching
              nature.oops(potato[iElk]);
              cntb++;                                             //Increment poached death counter
```

```
                        }

                        if((rndet(accRate))&&(potato[iElk].alive)){                         //Death due to
accidents
                            nature.oops(potato[iElk]);
                            cntc++;
                        }

                        if((rndet(unkRate))&&(potato[iElk].alive)){                         //Death due to
unknown reasons
                            nature.oops(potato[iElk]);
                            cntd++;
                        }

                        if((rndet(predRate))&&(potato[iElk].alive)){
                            nature.predator(50.00+50.00*Math.random(),potato[iElk]);
                            cnte++;
                        }

                        if(potato[iElk].alive){
                            potato[iElk].older();                           //Age the elk
                            potato[iElk].pregnancy(potato,Math.random());  //The random function to
determine pregnancy
                        }
                    }
                }
            }
            logData("Year.txt",nAlive);
        }
 //Reset the values for next run
        index = 0;
        month = 0;
        for(int c = 0; c< 65535; c++){    //Reset the container
            potato[c]=null;
        }
        potato[0] = new Elk();
        for(int j = 0; j < 25; j++){
            potato[0].birth(potato,index);
            potato[j] = new Elk();
        }
        sum = sum + nAlive;                //Statistical average of the number of elks alive
        sick = sick + cnta;
        accident = accident + cntc;
        unknown = unknown + cntd;
        poached = poached + cntb;
        predatorkilled = predatorkilled + cnte;
        logtData("Trial.txt",nAlive,cnta,cntc,cntd,cntb,cnte);
        logln("Year.txt");
        nAlive = 0;                        //Reset the alive elk counter at the end
        System.out.println(k);
        }
        System.out.println("Year: " + (year-1));
        oData(sum,sick,accident,unknown,poached,predatorkilled,agedeath);   //Output data
    }
}
```

b. Model 2: Model with sickness and predator rates as a function to time

```
/*
 * Calibrated Model B
 * Simulation trials: 0.5 Million
 * Simulation years: 2001-2026
 */
package himcm;
/**
 * @author Frank Gu and Andrew Tan
 */
import java.io.*;

public class HIMCM {
     static int nAlive = 0;                  //The amount of elks currently alive in the population
     static Elk[] potato = new Elk[65535];  //Create the set of Elk objects
     static int index = 0;                   //The index of the elks
     static int cCap = 1101;                 //Carrying Capacity
     static double OCDr = 1-1/cCap;          //Death probability due to over population
     static double pRate = 0.0396;           //Pregnancy rate
     static double sRate = 0.0;              //Sick rate (Modified in the sickness rate function)
     static double poaRate =0.0037/12;       //Poaching rate
     static double accRate = 0.0083/12;      //Accident rate
     static double unkRate = 0.0161/12;      //Unknown death rate
     static double predRate = 0.0287/12;     //Predator attack rate
     static int month = 0;                   //Month counter
     static int agedeath = 0;                //Amount of deaths due to old age
     static int k = 1;                       //Trials number counter
     static int year = 0;                    //Year counter
     static int trials = 500000;             //Number of times to run the simulation

public static boolean rndet(double limit) {         //A random function that throws back a boolean
    boolean rn = true;
    double rng = Math.random();
    if (rng>limit){
        rn = false;
    }
    return rn;
}
```

```
public static void logyData(String path,double pop){  //Prints data onto file(path, population)
    try{
        PrintWriter out = new PrintWriter(new BufferedWriter(new FileWriter(path, true)));
        out.println(year + "," + pop);
        out.close();
    }
    catch(IOException e){
        e.printStackTrace();
    }
}

public static void logtData(String path,double pop,int s,int a, int u, int p, int pk){  //Prints data
onto file(path, population, sickness,accident,unkonwn,poached,predator kills) by each trial
    try{
        PrintWriter out = new PrintWriter(new BufferedWriter(new FileWriter(path, true)));
        out.println(k + "," + pop + ","+ s + ","+ a + ","+ u + ","+ p + ","+ pk);
        out.close();
    }
    catch(IOException e){
        e.printStackTrace();
    }
}

public static void logln(String path){  //Prints data onto file(path, population,
sickness,accident,unkonwn,poached,predator,da) by each trial
    try{
        PrintWriter out = new PrintWriter(new BufferedWriter(new FileWriter(path, true)));
        out.println("");
        out.close();
    }
    catch(IOException e){
        e.printStackTrace();
    }
}

public static void lifecheck(Elk[] ek) {                    //Check the amount of Elks alive
    nAlive = 0;
    for (int i=0;i<index;i++){
        if (ek[i].alive = true){
            nAlive++;
        }
    }
}

public static void introduce(Elk[] e,int amount){
    for(int amt = 0; amt < amount; amt++){
        e[index] = new Elk();
        for(int i = 0; i < Math.random() * 15+5; i++){   //Give the generated elks a random age
between 5 and 20
            e[index].older();
        }
        index++;
    }
}

public static void oData(double pop, int s, int a,int u, int p, int pk,int da){             //Programming
debugging output
    System.out.println("Statistical Averages: ");
    System.out.println("Trials ran: " + (k-1));
    System.out.println("Population Alive: " + (int)(pop/k-3));                              //Minus 3 to
calibrate for population alive error
    System.out.println("Sickness: " + (int)(s/k-4));                                  //Minus 4 to calibrate
for sickness error
    System.out.println("Accidents: " + (int)a/k);
    System.out.println("Unknown: " + (int)u/k);
    System.out.println("Poached: " + (int)p/k);
    System.out.println("Predator Kill: " + (int)pk/k);
    System.out.println("Aged Death: " + (int)da/k);
}

public static void main(String[] args) {
    int sick = 0;
    int accident = 0;
    int unknown = 0;
    int poached = 0;
    int predatorkilled = 0;
    double sum = 0.0;
//---------------------------------------------------------------------------------------------------
    Elk[] potato = new Elk[65535];              //Create a new container for Elks
    introduce(potato,25);                       //Introduce 25 elks to the container
```

```
//----------------------------------------------------------------------------------------------------
  for(;k <= trials; k++){                    //Determine the number of times the cycle is run
    int cnta= 0;           //Sickness Death counter
    int cntb = 0;          //Poach Death counter
    int cntc  = 0;         //Accidental Death counter
    int cntd = 0;          //Unknown cause death counter
    int cnte = 0;          //Predator kill counter

    for(year = 2001; year <= 2056; year++){          //The number of years the sample endures
      if(year == 2002){                      //Introduce 27 elks in the second year
        introduce(potato,27);
      }
      if(year == 2007){                      //Introduce 5 elks in the 7th year
        introduce(potato,5);
      }

      sRate = 0.037*Math.log(year-2000)+0.1056;     //Logarithm to caliberate sicknessrate
with increasing population fitted with the data

      for(month = 1; month <= 12; month++){

        double natDead = nAlive * nAlive*0.16*(0.0009082652134);  //Calibrated populatino
suppressor, decimal is 1/1101
        boolean overCap = false;                    //Initialize overcapacity indicator
        lifecheck(potato);                       //Determine nAlive for set potato
        if(nAlive > cCap){                         //Determine whether nAlive is over the capcity
of the land
          overCap = true;
        }

        for(int iElk = 0; iElk < index; iElk++){          //The monthly cycle of one individual
elk begins here
          if((potato[iElk].alive)){                //First determine if the elk is alive or not
            if((overCap) && (rndet(OCDr))){            //Death with probability due to over
capcity
              potato[iElk].alive = false;
            }

            if((rndet(natDead/nAlive)) && (potato[iElk].alive)){
              potato[iElk].alive = false;
            }

            if((rndet(sRate))&&(potato[iElk].alive)){                    //Death due to
sickness
              potato[iElk].alive = false;
              if(!potato[iElk].alive){
                cnta++;   //Increment death due to sickness counter
              }
            }

            if((rndet(poaRate))&&(potato[iElk].alive)){                  //Death due to
poaching
              nature.oops(potato[iElk]);
              cntb++;                                      //Increment poached death counter
            }

            if((rndet(accRate))&&(potato[iElk].alive)){                  //Death due to
accidents
              nature.oops(potato[iElk]);
              cntc++;
            }

            if((rndet(unkRate))&&(potato[iElk].alive)){                  //Death due to
unknown reasons
              nature.oops(potato[iElk]);
              cntd++;
            }

            if((rndet(predRate))&&(potato[iElk].alive)){
              nature.predator(50.00+50.00*Math.random(),potato[iElk]);
              cnte++;
            }

            if(potato[iElk].alive){
              potato[iElk].older();                  //Age the elk
              potato[iElk].pregnancy(potato,Math.random());  //The random function to
determine pregnancy
            }
          }
        }
```

```
            }
            logyData("Year.txt",nAlive);                                  //Log yearly data
        }
 //Reset the values for next run
        index = 0;
        month = 0;
        for(int c = 0; c< 65535; c++){    //Reset the container
            potato[c]=null;
        }
        potato[0] = new Elk();
        for(int j = 0; j < 25; j++){
            potato[0].birth(potato,index);
            potato[j] = new Elk();
        }
       sum = sum + nAlive;                //Statistical average of the number of elks alive
       sick = sick + cnta;
       accident = accident + cntc;
       unknown = unknown + cntd;
       poached = poached + cntb;
       predatorkilled = predatorkilled + cnte;
       logtData("Trial.txt",nAlive,cnta,cntc,cntd,cntb,cnte);
       nAlive = 0;                        //Reset the alive elk counter at the end
       }
       System.out.println("Year: " + (year-1));
       oData(sum,sick,accident,unknown,poached,predatorkilled,agedeath);   //Output data
    }
}
```