

Simple Density-independent Growth

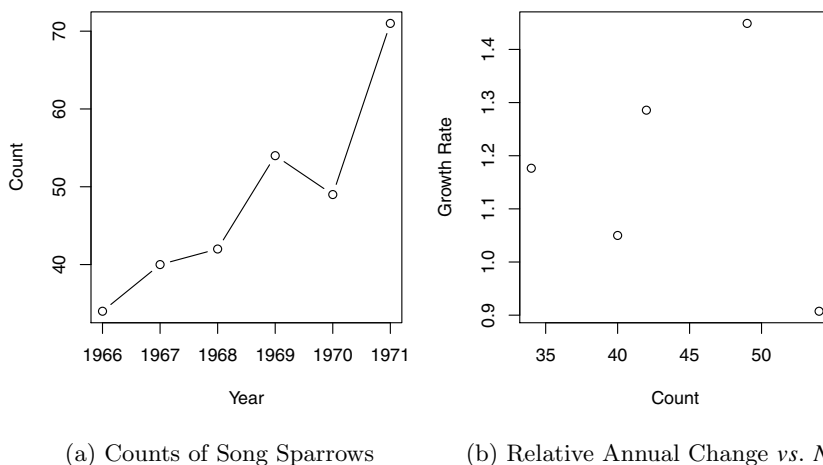


Fig. 1.1: Song Sparrow (*Melospiza melodia*) counts in Darrrtown, OH, USA. From Sauer, J. R., J. E. Hines, and J. Fallon. 2005. The North American Breeding Bird Survey, Results and Analysis 1966–2004. Version 2005.2. USGS Patuxent Wildlife Research Center, Laurel, MD.

Between 1966 and 1971, Song Sparrow (*Melospiza melodia*) abundance in Darrrtown, OH, USA, seemed to increase very quickly, seemingly unimpeded by any particular factor (Fig. 1.1a). In an effort to manage this population, we may want to predict its future population size. We may also want to describe its growth rate and population size in terms of mechanisms that could influence its growth rate. We may want to compare its growth and relevant mechanisms to those of other Song Sparrow populations or even to other passerine populations. These goals, *prediction*, *explanation*, and *generalization*, are frequently the goals toward which we strive in modeling anything, including populations, communi-

ties, and ecosystems. In this book, we start with simple models of populations and slowly add complexity to both the form of the model, and the analysis of its behavior. As we move along, we also practice applying these models to real populations.

What is a model, and why are they important in ecology? First, a model is an abstraction of reality. A road map, for instance, that you might use to find your way from Mumbai to Silvasaa is a model of the road network that allows you to predict which roads will get you to Silvasaa. As such, it *excludes* far more information about western India than it includes. Partly as a result of excluding this information, it is eminently useful for planning a trip. Models in ecology are similar. Ecological systems are potentially vastly more complex than just about any other system one can imagine for the simple reason that ecosystems are composed of vast numbers of genetically distinct individuals, each of which is composed of at least one cell (e.g., a bacterium), and all of these individuals may interact, at least indirectly. Ecological models are designed to capture particular key features of these potentially complex systems. The goal is to capture a key feature that is particularly interesting and useful.

In this book, we begin with the phenomenon called *density-independent growth*. We consider it at the beginning of the book for a few reasons. First, the fundamental process of reproduction (e.g., making seeds or babies) results in a *geometric series*.¹ For instance, one cell divides to make two, those two cells each divide to make four, and so on, where reproduction for each cell results in two cells, *regardless of how many other cells are in the population* — that is what we mean by *density-independent*. This myopically observed event of reproduction, whether one cell into two, or one plant producing many seeds, is the genesis of a geometric series. Therefore, most models of populations include this fundamental process of geometric increase. Second, populations can grow in a density-independent fashion when resources are plentiful. Third, it behooves us to start with this simple model because most, more complex population models include this process.

1.1 A Very Specific Definition

Density-independence in a real population is perhaps best defined quite specifically and operationally as a lack of a statistical relation between the density of a population, and its *per capita* growth rate. The power to detect a significant relation depends, in part, on those factors which govern power in statistical relations between any two continuous variables: the number of observations, and the range of the predictor variable. Therefore, our conclusion, that a particular population exhibits density-independent growth, may be trivial if our sample size is small (i.e., few generations sampled), or if we sampled the population over a very narrow range of densities. Nonetheless, it behooves us to come back

¹ A mathematical series is typically a list of numbers that follow a rule, and that you sum together.

to this definition if, or when, we get caught up in the biology of a particular organism.

We could examine directly the relation between the growth rate and population size of our Song Sparrow population (Fig. 1.1b). We see that there is no apparent relation between the growth rate and the density of the population.² That is what we mean by “density-independent growth.”

1.2 A Simple Example

Let’s pretend you own a small piece of property and on that property is a pond. Way back in June 2000, as a present for Mother’s Day, you were given a water lily (*Nymphaea odorata*), and you promptly planted it, with its single leaf or frond, in the shallows of your pond. The summer passes, and your lily blossoms, producing a beautiful white flower. The following June (2001) you notice that the lily grew back, and that there were three leaves, not just one. Perhaps you cannot discern whether the three leaves are separate plants. Regardless, the pond now seems to contain three times the amount of lily pad that it had last year.

The following June (2002) you are pleased to find that instead of three leaves, you now have nine. In June 2003, you have 27 leaves, and in 2004 you have 81 leaves (Fig. 1.3). How do we describe this pattern of growth? How do we predict the size of the population in the future? Can we take lessons learned from our water lily and apply it to white-tailed deer in suburbia, or to bacteria in the kitchen sink?

We rely on theory to understand and describe the growth of our water lily in such a way as to apply it to other populations. The role of theory, and theoretical ecology, is basically three-fold. We might like theory to allow us to describe the pattern in sufficient detail (1) to provide a *mechanistic explanation* for how the lily grew as fast or as slowly as it did, (2) allow us to make *predictions* about the population size in the future, and (3) allow us to *generalize* among other lily populations or other species. These goals typically compete with each other, so real models are mathematical descriptions that result from tradeoffs among these goals which depend precisely on our particular needs [109].

1.3 Exploring Population Growth

So, how fast are the lilies of the example growing? Between years 1 and 2, it increased by 2 fronds; between years 2 and 3, it increased by 6. In subsequent years it increased by 18, and 54 fronds. The number changes each year (Fig. 1.3), so how do we predict the future, or even explain the present? Can we find a general rule that works for any year?

² Consider that if area is fixed, “count” or population size differs from density by a fixed multiplier

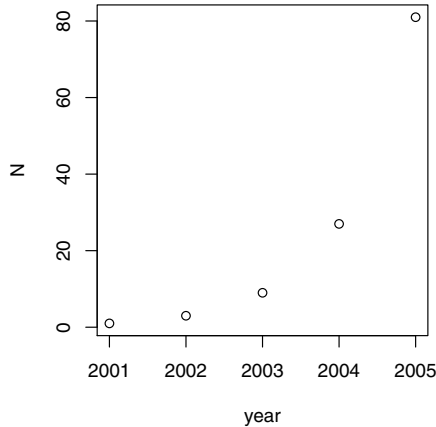


Fig. 1.2: Hypothetical water lily population size through time.

Simple Graphing of Population Size (Fig. 1.3)

Here we create two vectors: population size, N , and years. Using `c()` allows us to create an arbitrary vector, and the colon, `:`, provides a sequence of consecutive integers.

```
> N <- c(1, 3, 9, 27, 81)
> year <- 2001:2005
> plot(year, N)
```

The lily population (Fig. 1.3) increases by a different amount each year. What about proportions — does it increase by a different proportion each year? Let's divide each year's population size by the previous year's size, that is, perform N_{t+1}/N_t for all t , where t is any particular point in time, and $t + 1$ is the next point in time. For N , that amounts to $3/1$, $9/3$, \dots . What do we notice?

Vectorized math

Here we divide each element of one vector (the second through fifth element of N) by each element of another vector (the first through fourth elements of N).

```
> rates = N[2:5]/N[1:4]
> rates
[1] 3 3 3 3
```

Lo, and behold! all of these proportions are the same: 3. Every year, the lilies increase by the same proportion — they triple in abundance, increasing by 200%. That is the general rule that is specific to our population. It is general because it appears to apply to each year, and could potentially describe other populations; it is not, for instance, based on the photosynthetic rate in lily pads. It is specific because it describes a specific rate of increase for our water lily

population. We can represent this as

$$N_{2002} = 3 \times N_{2001}$$

where N_{2002} is the size of the population in 2002. If we rearrange this, dividing both sides by N_{2001} , we get

$$\frac{N_{2002}}{N_{2001}} = 3$$

where 3 is our rate of increase.

Generalizing this principle, we can state

$$N_{t+1} = 3N_t$$

$$\frac{N_{t+1}}{N_t} = 3.$$

1.3.1 Projecting population into the future

The above equations let us describe the rate of change population size N from one year to the next, but how do we predict the size 10 or 20 years hence? Let's start with one year and go from there.

$$N_{2002} = 3N_{2001}$$

$$N_{2003} = 3N_{2002} = 3(3N_{2001})$$

$$N_{2004} = 3N_{2003} = 3(3N_{2002}) = 3(3(2N_{2001}))$$

So, . . . what is the general rule that is emerging for predicting water lily N , some years hence? Recall that $3 \times 3 \times 3 = 3^3$ or $a \times a \times a = a^3$, so more generally, we like to state

$$N_t = \lambda^t N_0 \tag{1.1}$$

where t is the number of time units (years in our example), N_0 is the size of our population at the beginning, λ is the per capita rate of increase over the specified time interval and N_t is the predicted size of the population after t time units.

Note that lambda, λ , is the *finite rate of increase*. It is the per capita rate of growth of a population *if the population is growing geometrically*. We discuss some of the finer points of λ in Chapter 2. We can also define a related term, the *discrete growth factor*, r_d , where $\lambda = (1 + r_d)$.

Note that "time" is not in calendar years but rather in years since the initial time period. It is also the number of time steps. Imagine that someone sampled a population for five years, 1963–1967, then we have four time steps, and $t = 4$.

Projecting population size

Here we calculate population sizes for 10 time points beyond the initial. First we assign values for N_0 , λ , and time.

```
> N0 <- 1
> lambda <- 2
> time <- 0:10
```

Next we calculate N_t directly using our general formula.

```
> Nt <- N0 * lambda^time
> Nt
[1] 1 2 4 8 16 32 64 128 256 512 1024
```

1.3.2 Effects of initial population size

Let's explore the effects of initial population size. First, if we just examine equation 1.1, we will note that $N_t = N_0 \times \text{stuff}$. Therefore, if one population starts out twice as big as another, then it will always be twice as big, given geometric growth (Fig. 1.3a). We see that small initial differences diverge wildly over time (Fig. 1.3a), because “twice as big” just *looks* a lot bigger as the magnitude increases.

Effects of Initial Population Size

We first set up several different initial values, provide a fixed λ , and set times from zero to 4.

```
> N0 <- c(10, 20, 30)
> lambda <- 2
> time <- 0:4
```

We calculate population sizes at once using `sapply` to *apply* a function (`n*lambda^time`) to each element of the first argument (each element of N_0).

```
> Nt.s <- sapply(N0, function(n) n * lambda^time)
> Nt.s
```

```
      [,1] [,2] [,3]
[1,] 10 20 30
[2,] 20 40 60
[3,] 40 80 120
[4,] 80 160 240
[5,] 160 320 480
```

The result is a matrix, and we see N_0 in the first row, and each population is in its own column. Note that population 2 is always twice as big as population 1.

If we change the y-axis scale to logarithms, however, we see that the lines are parallel! Logarithms are a little weird, but they allow us to look at, and think

about, many processes where rates are involved, or where we are especially interested in the relative magnitudes of variables. Consider the old rule we get when we take the logarithm of both sides of an equation, where the right hand side is a ratio.

$$y = \frac{a}{b} \quad (1.2)$$

$$\log y = \log\left(\frac{a}{b}\right) = \log a - \log b \quad (1.3)$$

Thus, when we change everything into logarithms, ratios (like λ) become *differences*, which result in straight lines in graphs (Fig. 1.3b). On a linear scale, populations that are changing at the same *rates* can look very different (Fig. 1.3a), whereas on a logarithmic scale, the populations will have *parallel* trajectories (Fig. 1.3b).

Graphing a Matrix (Figs. 1.3a, 1.3b)

We can use `matplot` to plot a matrix vs. a single vector on the X-axis. By default it labels the points according to the number of the column

```
> matplot(time, Nt.s, pch = 1:3)
```

We can also plot it with a log scale on the y-axis.

```
> matplot(time, Nt.s, log = "y", pch = 1:3)
```

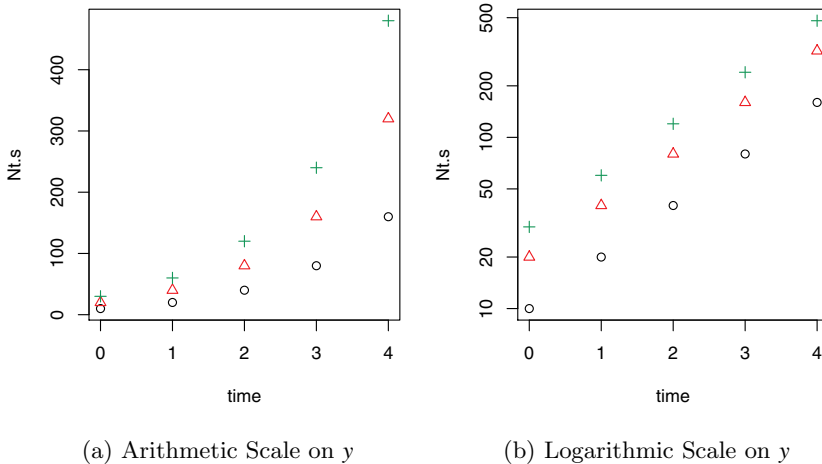


Fig. 1.3: Effects of variation in initial N on population size, through time. Different symbols indicate different populations.

Note that changing the initial population size changes the intercept. It also changes the slope in linear space, but not in log-linear space. It changes the absolute rate of increase ($N_2 - N_1$), but not the relative rate of increase (N_2/N_1).

1.3.3 Effects of different per capita growth rates

Perhaps the most important single thing we can say about λ is that when $\lambda < 1$, the population shrinks, and when $\lambda > 1$ the population grows. If we examine eq 1.1, $N_t = \lambda^t N_0$, we will note that λ is exponentiated, that is, raised to a power.³ It will always be true that when $\lambda > 1$ and $t > 1$, $\lambda^t > \lambda$. It will also be true that when $\lambda < 1$ and $t > 1$, $\lambda^t < \lambda$ (Fig. 1.4).

Thus we see the basis of a very simple but important truism. *When $\lambda > 1$, the population grows, and when $\lambda < 1$ the population shrinks* (Fig. 1.4). When $\lambda = 1$, the population size does not change, because $1^t = 1$ for all t .

Effects of Different λ (Fig. 1.4)

Here we demonstrate the effects on growth of $\lambda > 1$ and $\lambda < 1$. We set $N_0 = 100$, and time, and then pick three different λ .

```
> N0 <- 100
> time <- 0:3
> lambdas <- c(0.5, 1, 1.5)
```

We use `sapply` again to apply the geometric growth function to each λ . This time, `x` stands for each λ , which our function then uses to calculate population size. We then plot it, and add a reference line and a little text.

```
> N.all <- sapply(lambdas, function(x) N0 * x^time)
> matplot(time, N.all, xlab = "Years", ylab = "N", pch = 1:3)
> abline(h = N0, lty = 3)
> text(0.5, 250, expression(lambda > 1), cex = 1.2)
> text(0.5, 20, expression(lambda < 1), cex = 1.2)
```

The reference line is a *horizontal line* with the *line type* dotted. Our text simply indicates the regions of positive and negative growth.

We note that we have graphed *discrete* population growth. If we are counting bodies, and the population reproduces once per year, then the population will jump following all the births (or emergence from eggs). Further, it is probably always the case that following a bout of synchronous reproduction, we observe chronic ongoing mortality, with the result of population decline between spikes of reproduction. Nonetheless, unless we collect the data, we can't really say much about what goes on in between census periods.

1.3.4 Average growth rate

In any real data set, such as from a real population of *Nymphaea*, N_{t+1}/N_t will vary from year to year. Let's examine this with a new data set in which annual growth rate varies from year to year.

³ What happens to y^x as x increases, if $y > 1$ — does y^x increase? What happens if $y < 1$ — does y^x decrease? The answer to both these questions is yes.

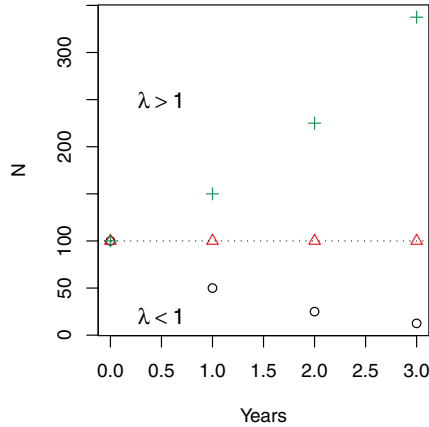


Fig. 1.4: Effects of variation in λ on population size through time. The dotted line indicates no change ($N_t = N_0$; $\lambda = 1$). Different symbols (circles, triangles, crosses) indicate populations resulting from $\lambda = (0.5, 1.0, 1.5)$, respectively. Any λ greater than 1 results in positive geometric growth; any $\lambda < 1$ results in negative geometric growth, or population decline.

Since growth rate varies from year to year, we may want to calculate average annual growth rate over several years. As we see below, however, the arithmetic averages are not really appropriate.

Consider that N_{t+1}/N_t may be a random variable which we will call R .⁴ That is, this ratio from any one particular year could take on a wide variety of values, from close to zero, up to some (unknown) large number. Let's pick two out of a hat, where $R = 0.5, 1.5$. The arithmetic average of these is 1.0, so this might seem to predict that, on average, the population does not change. Let's project the population for two years using each R once.

$$\begin{aligned} N_0 &= 100 \\ N_1 &= N_0 (0.5) = 50 \\ N_2 &= N_1 (1.5) = 75 \end{aligned}$$

We started with 100 individuals, but the population shrank! Why did that happen? It happens because, in essence, we *multiply* the λ together, where $N_2 = N_0 R_1 R_2$. In this case, then, what is a sensible “average”?

How do we calculate an average for things that we multiply together? We would like a value for R which would provide the solution to

$$\bar{R}^t = R_1 R_2 \dots R_t \quad (1.4)$$

⁴ Some authors use R for very specific purposes, much as one might use λ ; here we just use it for a convenient letter to represent observed per capita change.

where t is the number of time steps and R_1 is the observed finite rate of increase from year 1 to year 2. The bar over R indicates a mean.

All we have to do is solve for R .

$$(\bar{R}^t)^{1/t} = (R_1 R_2 \dots R_t)^{1/t} \quad (1.5)$$

$$\bar{R} = (R_1 R_2 \dots R_t)^{1/t} \quad (1.6)$$

$$(1.7)$$

We take the t -th root of the product of all the R . This is called the *geometric average*. Another way of writing this would be to use the product symbol, \prod , as in

$$\bar{R} = \left(\prod_{i=1}^t R_i \right)^{1/t} \quad (1.8)$$

If we examine the Song Sparrow data (Fig. 1.5), we see that projections based on the geometric average R are less than when based on the arithmetic average; this is always the case.

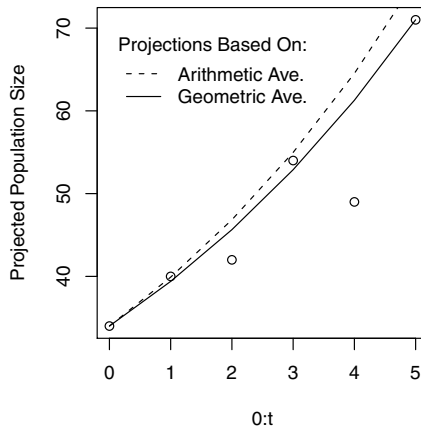


Fig. 1.5: Song Sparrow population sizes, and projections based on arithmetic and geometric mean R .

Comparing arithmetic and geometric averages (Fig. 1.5)

First we select the number of observed R ($t = 5$); this will require that we use six years of Song Sparrow data.

```
> t <- 5
> data(sparrows)
> SS6 <- sparrows[1:(t + 1), ]
```

Next we calculate λ for each generation, from t to $t + 1$, and calculate the arithmetic and geometric means.

```
> SSgr <- SS6$Count[2:(t + 1)]/SS6$Count[1:t]
> lam.A <- sum(SSgr)/t
> lam.G <- prod(SSgr)^(1/t)
```

Now we can plot the data, and the projections based on the two averages (Fig. 1.5).

```
> N0 <- SS6$Count[1]
> plot(0:t, SS6$Count, ylab = "Projected Population Size")
> lines(0:t, N0 * lam.A^(0:t), lty = 2)
> lines(0:t, N0 * lam.G^(0:t), lty = 1)
> legend(0,70,c("Arithmetic Ave.", "Geometric Ave."), title = "Projections Based On:",
+       lty = 2:1, bty = "n", xjust = 0)
```

1.4 Continuous Exponential Growth

Although many, many organisms are modeled well with discrete growth models (e.g., insects, plants, seasonally reproducing large mammals), many populations are poorly represented by discrete growth models. These populations (e.g., bacteria, humans) are often modeled as continuously growing populations. Such models take advantage of simple calculus, the mathematics of rates.

Whereas geometric growth is proportional change in a population over a specified finite time interval, exponential growth is proportional *instantaneous* change over, well, an instant.

Imagine a population of *Escherichia coli* started by inoculating fresh Luria-Bertania medium with a stab of *E. coli* culture. We start at time zero with about 1000 cells or CFUs (colony forming units), and wind up the next day with 10^{10} cells. If we used (incorrectly) a discrete growth model, we could calculate N_{t+1}/N_t and use this as an estimate for λ , where $\lambda = 10^{10}/10^3 = 10^7$ cells per cell per day. We know, however, that this is a pretty coarse level of understanding about the dynamics of this system. Each cell cycle is largely asynchronous with the others, and so many cells are dividing each second. We could simply define our time scale closer to the average generation time of a cell, for example $\lambda = 2$ cells cell⁻¹ 0.5 h⁻¹, but the resulting discrete steps in population growth would still be a poor representation of what is going on. Rather, we see population size changing very smoothly from hour to hour, minute to minute. Can we come up with a better description? Of course.

1.4.1 Motivating continuous exponential growth

If we assume that our *E. coli* cells are dividing asynchronously, then many cells are dividing each fraction of a second — we would like to make that fraction of a second an *infinitely* small time step. Unfortunately, that would mean that we have an infinitely large number of time steps between $t = 0$ and $t = 1$ day, and we couldn't solve anything.

A long time ago, a very smart person⁵ realized that geometric growth depends on how often you think a step of increase occurs. Imagine you think a population increases at an annual growth rate $\lambda = 1.5$. This represents a 50% increase or

$$N_1 = N_0 (1 + 0.5)$$

so the *discrete growth increment* is $r_d = 0.5$. You could reason that twice-annual reproduction would result in half of the annual r_d . You could then do growth over two time steps, and so we would then raise λ^2 , because the population is taking two, albeit smaller, time steps. Thus we would have

$$N_1 = N_0 (1 + 0.5/2)^2 = N_0 (1 + 0.25)^2$$

What if we kept increasing the number of time steps, and decreasing the growth increment? We could represent this as

$$N_1 = N_0 \left(1 + \frac{r_d}{n}\right)^n$$

$$\frac{N_1}{N_0} = \left(1 + \frac{r_d}{n}\right)^n$$

Our question then becomes, what is the value of $\left(1 + \frac{r_d}{n}\right)^n$ as n goes to infinity? In mathematics, we might state that we would like the solution to

$$\lim_{n \rightarrow \infty} \left(1 + \frac{r_d}{n}\right)^n. \quad (1.9)$$

To begin with, we simply try larger and larger values of n , graph eq. 1.9 *vs.* n , and look for a limit (Fig. 1.6).

⁵ Jacob Bernoulli (1654–1705)

Numerical approximation of e

Here we use brute force to try to get an approximate solution to eq. 1.9. We'll let n be the number of divisions within one year. This implies that the finite rate of increase during each of these fractional time steps is r_d/n . Let the $\lambda = 2$ and therefore $r_d = 1$. Note that because $N_0 = 1$, we could ignore it, but let's keep it in for completeness.

```
> n <- 0:100
> N0 <- 1
> rd <- 1
```

Next, we calculate $(1 + \frac{r_d}{n})^n$ for ever larger values of n .

```
> N1 <- N0 * (1 + rd/n)^n
```

Last, we plot the ratio and add some fancy math text to the plot (see `?plotmath` for details on mathematical typesetting in R).

```
> plot(n, N1/N0, type = "l")
> text(50, 2, "For n = 100,")
> text(50, 1.6, bquote((1 + frac("r"["d"], "n"))^"n" == .(round(N1[101]/N0,
+ 3))))
```

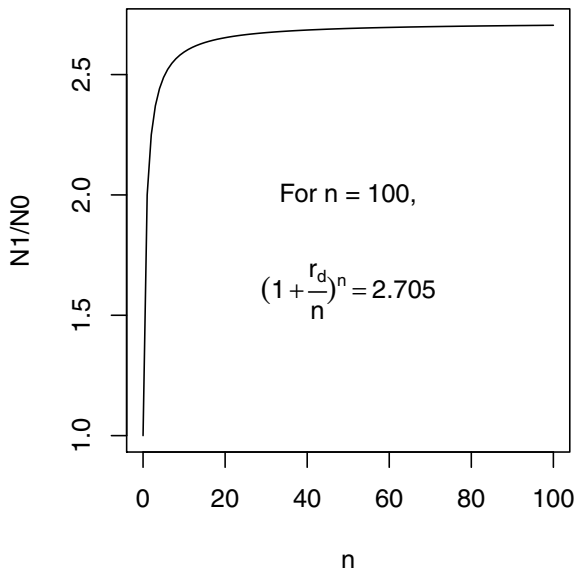


Fig. 1.6: The limit to subdividing reproduction into smaller steps. We can compare this numerical approximation to the true value, $e^1 = 2.718$.

Thus, when reproduction occurs continuously, the population can begin to add to itself right away. Indeed, if a population grew in a discrete annual step

$N_{t+1} = N_t(1 + r_d)$, the same r_d , divided up into many small increments, would result in a much larger increase.

It turns out that the increase has a simple mathematical expression, and we call it the *exponential*, e . As you probably recall, e is one of the magic numbers in mathematics that keeps popping up everywhere. In this context, we find that

$$\lim_{n \rightarrow \infty} \left(1 + \frac{r}{n}\right)^n = e^r \quad (1.10)$$

where e is the *exponential*.

This means that when a population grows geometrically, with infinitely small time steps, we say the population grows *exponentially*, and we represent that as,

$$N_t = N_0 e^{rt}. \quad (1.11)$$

We call r the *instantaneous* per capita growth rate, or the *intrinsic rate of increase*.

Projection of population size with continuous exponential growth is thus no more difficult than with discrete growth (Fig. 1.7).

Projecting a continuous population

We select five different values for r : two negative, zero, and two positive. We let t include the integers from 1 to 100. We then use `sapply` to *apply* our function of continuous exponential growth to each r , across all time steps. This results in a matrix where each row is the population size at each time t , and each column uses a different r .

```
> r <- c(-0.03, -0.02, 0, 0.02, 0.03)
> N0 <- 2
> t <- 1:100
> cont.mat <- sapply(r, function(ri) N0 * exp(ri * t))
```

Next we create side-by-side plots, using both arithmetic and logarithmic scales, and add a legend.

```
> layout(matrix(1:2, nrow = 1))
> matplot(t, cont.mat, type = "l", ylab = "N", col = 1)
> legend("topleft", paste(rev(r)), lty = 5:1, col = 1, bty = "n",
+       title = "r")
> matplot(t, cont.mat, type = "l", ylab = "N", log = "y", col = 1)
```

1.4.2 Deriving the time derivative

We can also differentiate eq. 1.11 with respect to time to get the differential equation for instantaneous population growth rate. Recall that the chain rule tells us that the derivative of a product of two terms is the sum of the products of the derivative of one times the other original term.

$$\frac{d}{dt}(XY) = \frac{dX}{dt}Y + \frac{dY}{dt}X$$

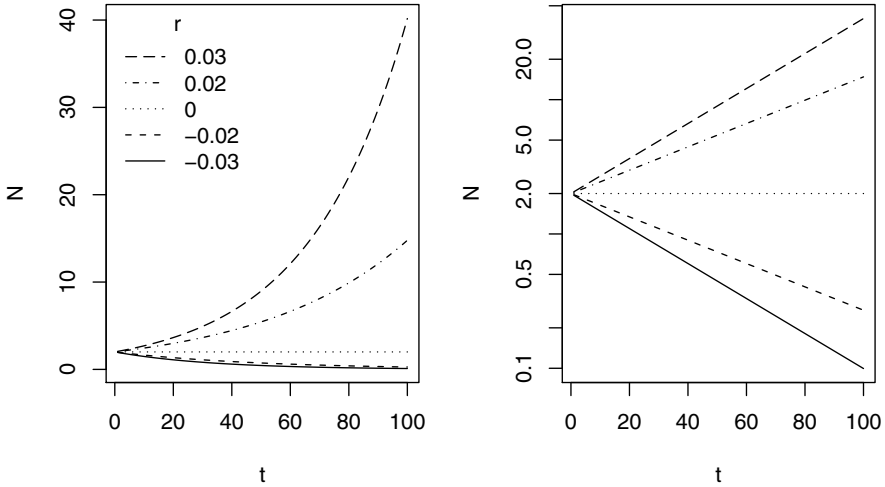


Fig. 1.7: Projecting continuous populations with different r .

Therefore to begin to differentiate eq. 1.11, with respect to t , we have,

$$\frac{d}{dt} N_0 e^{rt} = \frac{d}{dt} N_0 \cdot (e^r)^t + \frac{d}{dt} (e^r)^t \cdot N_0$$

Recall also that the derivative of a constant is zero, and the derivative of a^t is $\ln(a) \cdot a^t$, resulting in,

$$\frac{d}{dt} N_0 e^{rt} = 0 \cdot (e^r)^t + \ln e^r \cdot (e^r)^t \cdot N_0$$

Given that $\ln e = 1$, and that $N_0 e^{rt} = N$ for any time t , this reduces to eq. 1.12. The time derivative, or differential equation, for exponential growth is

$$\frac{dN}{dt} = rN. \tag{1.12}$$

1.4.3 Doubling (and tripling) time

For heuristic purposes, it is frequently nice to express differences in growth rates in terms of something more tangible than a dimensionless number like r . It is relatively concrete to say population X increases by about 10% each year ($\lambda = 1.10$), but another way to describe the rate of change of a population is to state the amount of *time* associated with a dramatic but comprehensible change. The *doubling time* of a population is the time required for a population to double in size. Shorter doubling times therefore mean more rapid growth.

We could determine doubling time graphically. If we examine the expanding population in Fig. 1.4, we see that it takes about one and half years for the

population size to change from $N = 100$ to $N = 200$. Not surprisingly, we can do better than that. By doubling, we mean that $N_t = 2N_0$. To get the time at which this occurs, we solve eq. (1.11) for t ,

$$2N_0 = N_0e^{rt} \quad (1.13)$$

$$2 = e^{rt} \quad (1.14)$$

$$\ln(2) = rt \ln(e) \quad (1.15)$$

$$t = \frac{\ln(2)}{r}. \quad (1.16)$$

Thus, eq. 1.16 gives us the time required for a population to double, given a particular r . We could also get any arbitrary multiple m of any arbitrary initial N_0 .

Creating a function for doubling time

We can create a *function* for this formula, and then evaluate it for different values of m and r . For $m = 2$, we refer to this as “doubling time.” When we define the function and include arguments r and m , we also set a default value for $m=2$. This way, we do not always have to type a value for m ; by default the function will return the doubling time.

```
> m.time <- function(r, m = 2) {
+   log(m)/r
+ }
```

Now we create a vector of r , and then use `m.time` to generate a vector of doubling times.

```
> rs <- c(0, 1, 2)
> m.time(rs)
```

```
[1] Inf 0.6931 0.3466
```

Note that R tells us that when $r = 0$, it takes an infinite (`Inf`) amount of time to double. This is what we get when we try to divide by zero!

1.4.4 Relating λ and r

If we assume constant exponential and geometric growth, we can calculate r from data as easily as λ . Note that, so rearranging, we see that

$$N_t = N_0e^{rt}$$

$$\ln(N_t) = \ln(N_0) + rt.$$

In other words, r is the slope of the linear relation between $\ln(N_t)$ and time (Fig. 1.7), and $\ln(N_0)$ is the y-intercept. If the data make sense to fit a straight regression line to log-transformed data, the slope of that line would be r .

It also is plain that,

$$\lambda = e^r \quad (1.17)$$

$$\ln \lambda = r. \quad (1.18)$$

Summarizing some of what we know about how λ and r relate to population growth:

No Change $\lambda = 1, r = 0$

Population Growth $\lambda > 1, r > 0$

Population Decline $\lambda < 1, r < 0$

Remember — λ is for populations with discrete generations, and r is for continuously reproducing populations.

Units

What are the units for λ and r ? As λ is a ratio of two population sizes, the units could be individuals/individual, thus rendering λ dimensionless. Similarly, we can view λ as the net number of individuals produced by individuals in the population such that the units are net new individuals per individual per time step, or $\text{inds ind}^{-1} t^{-1}$. The intrinsic rate of increase, r , is also in units of $\text{inds ind}^{-1} t^{-1}$.

Converting between time units

A nice feature of r as opposed to λ is that r can be used to scale easily among time units. Thus, $r = 0.1 \text{ inds ind}^{-1} \text{ year}^{-1}$ becomes $r = 0.1/365 = 0.00027 \text{ inds ind}^{-1} \text{ day}^{-1}$. *You cannot do this with λ .* If you would like to scale λ from one time unit to another, first convert it to r using logarithms, make the conversion, then convert back to λ .

1.5 Comments on Simple Density-independent Growth Models

It almost goes without saying that if we are considering density-independent growth models to be representative of real populations, we might feel as though we are making a very long list of unrealistic assumptions. These might include no immigration or emigration, no population structure (i.e. all individuals are identical), and you can probably come up with many others [58]. However, I would argue vociferously that we are making only one assumption:

N increases by a constant per capita rate over the time interval(s) of interest.

Think about that. I am *not* saying that competition is not occurring, or that no death is occurring, or that all individuals are reproductively viable, or there is no abiotic disturbance, or that there is no population genetic structure. I am just saying that for the time period of interest, all things balance out, or are

of no substantive consequence, and the population chugs along at a particular pace.

If the per capita rate is constant, then there can be no statistical relation between the size of the population and its per capita growth rate. *In the absence of such a relation*, we say that the growth rate is density-independent.

Other ecologists will disagree with my sentiments regarding an absence of assumptions. That's OK — still others may agree with these sentiments. Take it upon yourself to acquire multiple perspectives and evaluate them yourself.

Both λ and r obviously depend on birth rates and death rates. For instance, we could view geometric growth as

$$N_{t+1} = N_t + BN_t - DN_t \quad (1.19)$$

where B is the number of births per individual and D is the probability of an individual dying during the specified time interval. Lambda, in this case, is $1 + (B - D)$ and $r_d = B - D$. This form would be nice if we had data on births and deaths, because, after all, one goal of Science is to explain complex phenomena (e.g., λ) in terms of their determinants (e.g., B and D). Similarly, we can state $r = b - d$ where b and d are per capita instantaneous rates. Such an advance in understanding the determinants would be great.

Perhaps now is a good time to think about all the assumptions others might tell us we are making when we present the above formulation. Are all individuals in the population equally likely to produce progeny and/or die? Will birth and death rates vary over time or as the size of the population changes more? How will resource supply rate influence these rates? Is there really no immigration or emigration? These are very interesting questions.

Simple density-independent growth provides, in some sense, a null hypothesis for the dynamic behavior of a population. Malthus warned us that organisms produce more progeny than merely replacement value, and population growth is an exponential (or geometric) process [125]. The question then becomes “What causes population growth to differ from a constant rate of change?” That, in a nutshell, is what the rest of the book is about.

1.6 Modeling with Data: Simulated Dynamics

The main purpose of this section⁶ is to begin to understand the mechanics of simulation. The preceding sections (the bulk of the chapter) emphasized understanding the deterministic underpinnings of simple forms of density independent growth: geometric and exponential growth. This section explores the simulation of density independent growth.

When we model populations, perhaps to predict the size of a population in the future, we can take a variety of approaches. One type of approach emphasizes deterministic prediction, using, for instance, \bar{R} . Another approach is to *simulate* population dynamics and we take this up in this next section.

⁶ This section emphasizes work in R.

To project population growth into the future should include some quantification of the uncertainty with our guess. Simulation is one way we can project populations and quantify the uncertainty. The way one often does that is to use the original data and sample it randomly to calculate model parameters. In this fashion, the simulations are random, but based on our best available knowledge, i.e., the real data. The re-use of observed data occurs in many guises, and it is known generally as bootstrapping or resampling.

1.6.1 Data-based approaches

In using our data to predict population sizes, let us think about three levels of biological organization and mechanism: population counts, changes in population counts, and individual birth and death probabilities. First, our count data alone provide a sample of a very large number of different possible counts. If we assume that there will be no trend over time, then a simple description of our observed counts (e.g., mean and confidence intervals) provide all we need. We can say “Song Sparrow counts in the Breeding Bird Survey in Darrrtown, OH, are about 51.”

Second, we could use the observed *changes* in population counts $R_t = N_{t+1}/N_t$ as our data. We would then draw an R_t at random from among the many observed values, and project the population one year forward. We then repeat this into the future, say, for ten years. Each simulation of a ten year period will result in a different ten year trajectory because we draw R_t at random from among the observed R_t . However, if we do many such simulations, we will have a *distribution* of outcomes that we can describe with simple statistics (e.g., median, mean, quantiles).

Third, we might be able to estimate the individual probabilities of births and deaths in the entire Darrrtown population, and use those probabilities and birth rates to simulate the entire population into the future. In such an *individual-based* simulation, we would simulate the fates of individuals, keeping track of all individual births and deaths.

There are myriad others approaches, but these give you a taste of what might be possible. In this section we focus on the second of these alternatives, in which we use observed R_t to simulate the dynamics of Song Sparrow counts.

Here we investigate Song Sparrow (*Melospize melodia*) dynamics using data from the annual U.S. Breeding Bird Survey (<http://www.mbr-pwrc.usgs.gov/bbs/>). Below we will

1. look at and collecting the data (annual R 's),
2. simulate one projection,
3. scale up to multiple simulations,
4. simplify simulations and perform 1000's, and
5. analyze the output.

1.6.2 Looking at and collecting the data

Let's start by looking at the data. *Looking at the data is always a good idea — it is a principle of working with data.* We first load the data from the **primer**

R package, and look at the names of the data frame. We then choose to `attach` the data frame, because it makes the code easier to read.⁷

```
> names(sparrows)
[1] "Year"          "Count"         "ObserverNumber"
> attach(sparrows)
```

Now we plot these counts through time (Fig. 1.8).

```
> plot(Count ~ Year, type = "b")
```

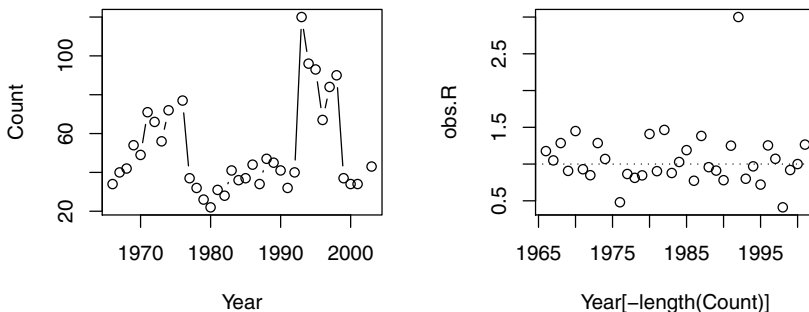


Fig. 1.8: Observations of Song Sparrows in Darrrtown, OH (<http://www.mbr-pwrc.usgs.gov/bbs/>).

We see that Song Sparrow counts⁸ at this site (the DARRTOWN transect, OH, USA) fluctuated a fair bit between 1966 and 2003. They never were completely absent and never exceeded ~ 120 individuals.

Next we calculate annual $R_t = N_{t+1}/N_t$, that is, the observed growth rate for each year t .⁹

```
> obs.R <- Count[-1]/Count[-length(Count)]
```

Thus our *data* are the observed R_t , not the counts *per se*. These R form the basis of everything else we do. Because they are so important, let's plot these as well. Let's also indicate $R = 1$ with a horizontal dotted line as a visual cue

⁷ I typically do not use `attach` but rather choose to always define explicitly the parent data frame I am using. It helps me reduce silly mistakes.

⁸ Recall that these are *samples* or observations of sparrows. These are *not* population sizes. Therefore, we will be simulating sparrows counts, not sparrow population sizes.

⁹ The use of “-” in the index tells R to exclude that element (e.g., `-1` means “exclude the first element of the vector”).

for zero population growth. Note that we exclude the last year because each R_t is associated with N_t rather than N_{t+1} .

```
> plot(obs.R ~ Year[-length(Count)])
> abline(h = 1, lty = 3)
```

One thing that emerges in our graphic data display (Fig. 1.8) is we have an unusually high growth rate in the early 1990's, with the rest of the data clustered around 0.5–1.5. We may want to remember that.

1.6.3 One simulation

Now that we have our randomly drawn R s, we are ready to simulate dynamics. A key assumption we will make is that *these R are representative of R in the future, and that each is equally likely to occur*. We then *resample* these observed R with *replacement* for each year of the simulation. This random draw of observed R 's then determines one random realization of a possible population trajectory. Let's begin.

First, we decide how many years we want to simulate growth.

```
> years <- 50
```

This will result in 51 population sizes, because we have the starting year, year 0, and the last year.

Next we draw 50 R at random with replacement. This is just like having all 35 observed R written down on slips of paper and dropped into a paper bag. We then draw one slip of paper out of the bag, write the number down, and put the slip of paper back in the bag, and then repeat this 49 more times. This is *resampling with replacement*¹⁰. The R function `sample` will do this. Because this is a pseudorandom¹¹ process, we use `set.seed` to make your process the same as mine, i.e., repeatable.

```
> set.seed(3)
> sim.Rs <- sample(x = obs.R, size = years, replace = TRUE)
```

Now that we have these 50 R , all we have to do is use them to determine the population size through time. For this, we need to use what programmers call a *for-loop* (see B.6 for further details). In brief, a for-loop repeatedly *loops* through a particular process, with one loop *for* each value of some indicator variable. Here we calculate each sparrow count in the next year, N_{t+1} , using the count in the current year N_t and the randomly drawn R *for* each year t .

¹⁰ We could resample *without* replacement. In that case, we would be assuming that all of these R_t are important and *will* occur at some point, but we just don't know when — they constitute the entire universe of possibilities. Sampling *with* replacement, as we do above, assumes that the observed R_t are all equally likely, but none is particularly important — they are just a sample of what is possible, and they might be observed again, or they might not.

¹¹ A *pseudorandom* process is the best computers can do — it is a complex deterministic process that generates results that are indistinguishable from random.

We begin by creating an empty output vector that is the right length to hold our projection, which will be the number of R s plus one.¹²

```
> output <- numeric(years + 1)
```

We want to start the projection with the sparrow count we had in the last year (the “maximum,” or biggest, year) of our census data.

```
> output[1] <- Count[Year == max(Year)]
```

Now the fun really starts to take off, as we finally use the for-loop. For each year t , we multiply N_t by the randomly selected R_t to get N_{t+1} and put it into the $t + 1$ element of `output`.

```
> for (t in 1:years) output[t + 1] <- {
+   output[t] * sim.Rs[t]
+ }
```

Let’s graph the result.

```
> plot(0:years, output, type = "l")
```

It appears to work (Fig. 1.9a) — at least it did something! Let’s review what we have done. We

- had a bird count each year for 36 years. From this we calculated 35 R (for all years except the very last).
- decided how many years we wanted to project the population (50 y).
- drew *at random and with replacement* the observed R — one R for each year we want to project.
- got ready to do a simulation with a for-loop — we created an empty vector and put in an initial value (the last year’s real data).
- performed each year’s calculation, and put it into the vector we made.

So what does Fig. 1.9a represent? It represents one possible outcome of a trajectory, if we assume that R has an equal probability of being any of the observed R_t . This *particular* trajectory is very unlikely, because it would require one particular sequence of R s. However, our simulation assumes that it is *no less likely* than any other particular trajectory.

As only one realization of a set of randomly selected R , Fig. 1.9a tells us very little. What we need to do now is to replicate this process a very large number of times, and examine the *distribution* of outcomes, including moments of the distribution such as the mean, median, and confidence interval of eventual outcomes.

1.6.4 Multiple simulations

Now we create a way to perform the above simulation several times. There are a couple tricks we use to do this. We still want to start small so we can figure out the steps as we go. Here is what we would do next.

¹² Remember that we always have one more population count than we do R_t .

- We start by specifying that we want to do 10 simulations, where one simulation is what we did above.
- We will need to use $50 \times 10 = 500$ randomly drawn R s and store those in a matrix.
- To do the ten separate, independent simulations, we will use `sapply`, to “apply” our simulations ten times. We have to use a for-loop for each population simulation, because each N_t depends on the previous N_{t-1} . We use `sapply` and related functions for when we want to do more than one independent operation.

Here we specify 10 simulations, create a matrix of the 10×50 randomly drawn R .

```
> sims = 10
> sim.RM <- matrix(sample(obs.R, sims * years, replace = TRUE),
+   nrow = years, ncol = sims)
```

Next we get ready to do the simulations. First, to hold each projection temporarily, we will reuse `output` as many times as required. We then *apply* our for-loop projection as many times as desired, for each value of `1:sims`.

```
> output[1] <- Count[Year == max(Year)]
> outmat <- sapply(1:sims, function(i) {
+   for (t in 1:years) output[t + 1] <- output[t] * sim.RM[t,
+     i]
+   output
+ })
```

Now let’s peek at the results (Fig. 1.9b). This is fun, but also makes sure we are not making a heinous mistake in our code. Note we use log scale to help us see the small populations.

```
> matplot(0:years, outmat, type = "l", log = "y")
```

What does it mean that the simulation has an approximately even distribution of final population sizes *on the log scale* (Fig. 1.9b)? If we plotted it on a linear scale, what would it look like?¹³

Rerunning this simulation, with new R each time, will show different dynamics every time, and that is the point of simulations. Simulations are a way to make a few key assumptions, and then leave the rest to chance. In that sense it is a null model of population dynamics.

¹³ Plotting it on the log scale reveals that the relative change is independent of population size; this is true because the rate of change is geometric. If we plotted it on a linear scale, we would see that many trajectories result in small counts, and only a few get really big. That is, the median size is pretty small, but a few populations get huge.

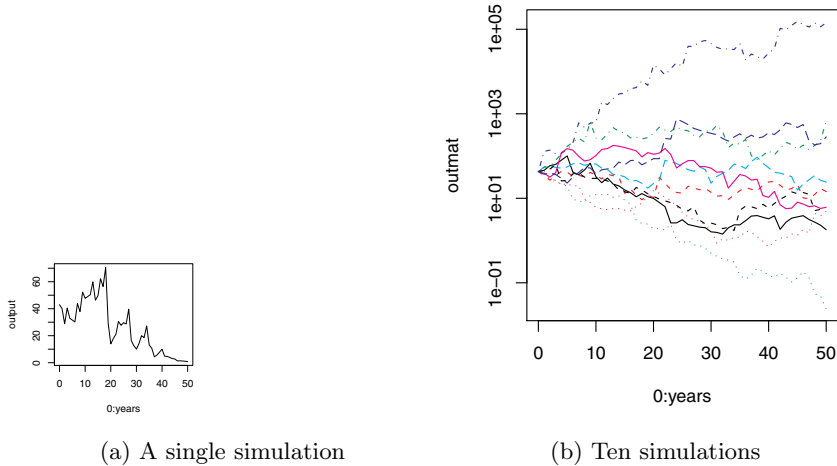


Fig. 1.9: Simulated population dynamics with R drawn randomly from observed Song Sparrow counts.

1.6.5 Many simulations, with a function

Let's turn our simulation into a user-defined function¹⁴ that simplifies our lives. We also add another assumption: individuals are irreducible. Therefore, let us use `round(,0)` to round to zero decimal places, i.e., the nearest integer.¹⁵

Our user defined function, `PopSim`, simply wraps the previous steps up in a single function.¹⁶ The output is a matrix, like the one we plotted above.

```
> PopSim <- function(Rs, N0, years = 50, sims = 10) {
+   sim.RM = matrix(sample(Rs, size = sims * years, replace = TRUE),
+     nrow = years, ncol = sims)
+   output <- numeric(years + 1)
+   output[1] <- N0
+   outmat <- sapply(1:sims, function(i) {
+     for (t in 1:years) output[t + 1] <- round(output[t] *
+       sim.RM[t, i], 0)
+     output
+   })
+   return(outmat)
+ }
```

If you like, try to figure out what each step of the simulation is doing. Consider it one of the end-of-chapter problems. Rely on on the code above to help you decipher the function.

¹⁴ For user-defined functions, see sec. B.4.1.

¹⁵ We could use also use `floor` to round down to the lowest integer, or `ceiling` to round up.

¹⁶ This process, of working through the steps one at a time, and then wrapping up the steps into a function, is a useful work flow.

Now we have the pleasure of using this population simulator to examine a number of things, including the sizes of the populations after 50 years. I first simulate 1000 populations,¹⁷ and use `system.time` to tell me how long it takes on my computer.

```
> system.time(output <- PopSim(Rs = obs.R, NO = 43, sims = 1000))

  user  system elapsed 
0.404  0.004  0.407
```

This tells me that it took less than half a second to complete 1000 simulations. That helps me understand how long 100 000 simulations might take. We also check the dimensions of the output, and they make sense.

```
> dim(output)

[1] 51 1000
```

We see that we have an object that is the size we think it should be. We shall assume that everything worked way we think it should.

1.6.6 Analyzing results

We extract the last year of the simulations (last row), and summarize it.

```
> N.2053 <- output[51, ]
> summary(N.2053, digits = 6)

  Min.  1st Qu.  Median    Mean  3rd Qu.    Max. 
  0.0    14.0    66.0   1124.6  291.8 332236.0
```

We see from this summary that the median final population size, among the 1000 simulations, is 66 individuals (median=50% quantile). While at least one of the populations has become extinct (min. = 0), the maximum is huge (max. = 332236). The `quantile` function allows us to find a form of empirical confidence intervals, including, approximately, the central 95% of the observations.¹⁸

```
> quantile(N.2053, prob = c(0.0275, 0.975))

2.75% 97.5%
 0 5967
```

These quantiles suggest that in 2053, we might observe sparrow counts anywhere from 0 to 5967, where zero and ~ 6000 are equally likely.

Notice the huge difference between the mean, $N = 1125$, and the median, $N=66$. Try to picture a histogram for which this would be true. It would be skewed right (long right hand tail), as with the lognormal distribution; this is common in ecological data.

¹⁷ If we were doing this in a serious manner, we might do 10–100 000 times.

¹⁸ Note that there are many ways to estimate quantiles (R has nine ways), but they are approximately similar to percentiles.

Let's make a histogram of these data. Exponentially growing populations, like this one, frequently show a lognormal distribution of abundances. Indeed, some say the “natural” unit of a population is $\log(N)$, rather than N . We will plot two frequency distributions of the final populations, one on the original scale, one using the logarithms of the final population sizes plus 1 (we use $N + 1$ so that we can include 0's — what is $\log(0)$? $\log(1)$?).

```
> hist(N.2053, main = "N")
> hist(log10(N.2053 + 1), main = "log(N+1)")
> abline(v = log10(quantile(N.2053, prob = c(0.0275, 0.975)) +
+ 1), lty = 3)
```

We added some reference lines on the second histogram, showing the 2.5 and 97.5% quantiles (Fig. 1.10). You can see that the logarithms of the population sizes are much more well-behaved, more symmetrical.

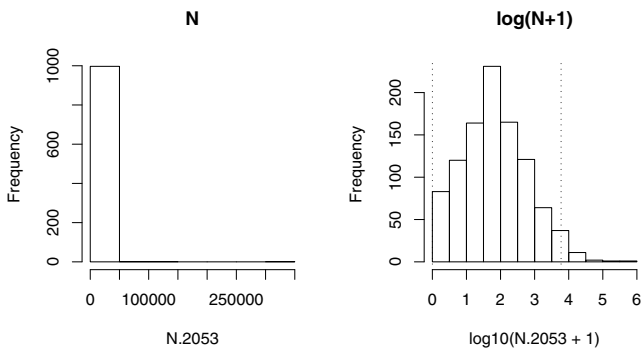


Fig. 1.10: Exploratory graphs of the distributions of the final simulated population sizes.

Can we really believe this output? To what can we compare our output? One thing that occurs to me is to compare it to the lower and upper bounds that we might contrive from *deterministic* projections.

To compare the simulation to deterministic projections, we could find the 95% t -distribution based confidence limits for the geometric mean of R . If we use our rules regarding the geometric mean, we would find that the logarithm of the *geometric* mean of R is the arithmetic mean of the $\log R$. So, one thing we could do is calculate the t -based confidence limits¹⁹ on $\log R$, backtransform these to project the population out to 2053 with lower and upper bounds.

Here we take find the logarithms, calculate degrees of freedom and the relevant quantiles for the t distribution.

¹⁹ Remember: the t -distribution needs the degrees of freedom, and a 95% confidence region goes from the 2.5% and the 97.5% percentiles.

```
> logOR <- log(obs.R)
> n <- length(logOR)
> t.quantiles <- qt(c(0.025, 0.975), df = n - 1)
```

Next we calculate the standard error of the mean, and the 95% confidence limits for $\log R$.

```
> se <- sqrt(var(logOR)/n)
> CLs95 <- mean(logOR) + t.quantiles * se
```

We backtransform to get R , and get a vector of length 2.

```
> R.limits <- exp(CLs95)
> R.limits
```

```
[1] 0.8968 1.1302
```

What do we see immediately about these values? One is less than 0, and one is greater than 0. This means that for the lower limit, the population will shrink (geometrically), while for the upper limit, the population will increase (geometrically). Let's go ahead and make the 50 y projection.

```
> N.Final.95 <- Count[Year == max(Year)] * R.limits^50
> round(N.Final.95)
```

```
[1] 0 19528
```

Here we see that the lower bound for the deterministic projection is the same (extinction) as the simulation, while the upper bound is much greater than that for the simulation. Why would that be? Perhaps we should examine the assumptions of our deterministic approach.

We started by assuming that the $\log R$ could be approximated with the t distribution, one of the most pervasive distributions in statistics and life. Let's check that assumption. We will compare the $\log R$ to the theoretical values for a t distribution. We scale $\log OR$ to make the comparison more clear.

```
> qqplot(qt(ppoints(n), df = n - 1), scale(logOR))
> qqline(scale(logOR))
```

How do we interpret these results? If the distribution of an observed variable is consistent with a particular theoretical distribution, the ordered quantiles of data will be a linear (straight line) function of the theoretical quantiles of the theoretical distribution. Deviations from that straight line illustrate how the data deviate. Here we see that the data have three outliers that are much more extreme (greater and smaller) than expected in the t -distribution, and more data are cluster around the center of the distribution than we would expect. We should ask whether those extreme values are mistakes in data collection or recording or whether they are every bit as accurate as all the other measurements.

Compare our two different confidence limits. These provide two different answers to our original question, "what might be the Song Sparrow count at this site in 2053?" Both of these assume a similar underlying model, density

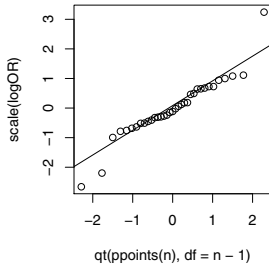


Fig. 1.11: Quantile-quantile plot used to compare $\log R$ to a t -distribution. Scaling $\log OR$ in this case means that we subtracted the mean and divided by the standard deviation. A histogram performs a similar service but is generally less discriminating and informative.

independent growth, but give different answers. Of which approach are we more confident? Why? What assumptions does each make?

We can be quite sure that our assumption regarding the t -distribution of our R is unsupported — our data have outliers, relative to a t -distribution. What would this do? It would increase the variance of our presumed distribution, and lead to wider confidence intervals, even though most of the data conform to a narrower distribution. Our simulation procedure, on the other hand, rarely samples those extreme points and, by chance, samples observed R that fall much closer to the median. This can occasionally be a problem in simulations based on too little data — the data themselves do not contain enough variability. Imagine the absurdity of a data-based simulation that relies on one observation — it would be *very* precise (but wrong)!

Our conclusions are based on a model of discrete density-independent population growth — what assumptions are we making? are they valid? Are our unrealistic assumptions perhaps nonetheless a good approximation of reality? We will revisit these data later in the book (Chapter 3) to examine one of these assumptions. We do not need to answer these questions now, but it is essential, and fun, to speculate.

1.7 Summary

In this chapter, we have explored the meaning of density-independent population growth. It is a statistically demonstrable phenomenon, wherein the per capita growth rate exhibits no relation with population density. It is a useful starting point for conceptualizing population growth. We have derived discrete geometric and continuous exponential growth and seen how they are related. We have calculated doubling times. We have discussed the assumptions that

different people might make regarding these growth models. Last, we have used simulation to explore prediction and inference in a density-independent context.

Problems

1.1. Geometric growth Analyze the following data, relying on selected snippets of previous code.

- In the years 1996 through 2005, lily population sizes are $N = 150, 100, 125, 200, 225, 150, 100, 175, 100, 150$. Make a graph of population size versus time.
- Calculate R for each year; graph R vs. time.
- Calculate arithmetic and geometric average growth rates of this population.
- Based on the appropriate average growth rate, what would be the expected population size in 2025? What would the estimated population size be if you used the inappropriate mean? Do not use simulation for this.
- (d*) Given these data, develop simulations as above with the user-defined function, `PopSim`. Describe the distribution of projected population sizes for 2010.

1.2. Doubling Time

- Derive the formula for doubling time in a population with continuous exponential growth.
- What is the formula for tripling time?
- If we are modeling humans or *E. coli*, would a model of geometric, or exponential growth be better? Why?
- If an *E. coli* population grew from 1000 cells to 2×10^9 cells in 6 h, what would its intrinsic rate of increase be? Its doubling time?

1.3. Human Population Growth

- There were about 630 million people on the planet in 1700, and 6.3 billion in 2003 [33]. What was the intrinsic rate of increase, r ?
- Graph the model of human population size population size from 1700 to 2020.
- Add points on the graph indicating the population doublings from 1700 onward.
- (d*) What will prevent humans from outweighing the planet by the end of this century? What controls human population growth? Do these controls vary spatially across the planet? See Cohen [33] to get going.

1.4. R functions

Find the R functions in Chapter 1. Demonstrate their uses.