

Ecología de poblaciones silvestres

David Martínez Cascante

2018-05-23

Índice general

1. Introducción	5
2. Modelos de crecimiento	7
2.1. Crecimiento denso-independiente	8
2.2. Crecimiento denso-dependiente	14
2.3. Otras fuentes bibliográficas	20
3. Demografía	21
3.1. Tablas de vida	23
3.2. Curvas de supervivencia	27
3.3. Análisis de Curvas de Captura	29
3.4. Matrices de transición	33
3.5. Otras fuentes bibliográficas	45
4. Estimación de abundancia	47
4.1. Abundancia no es un conteo	47
5. Interacciones entre poblaciones	59
6. Soluciones a los ejercicios	65
A. Métodos numéricos para ecología de poblaciones	71
A.1. Simulación de ecuaciones diferenciales	71
A.2. Error en las simulaciones	76
B. Tutorial de R con RStudio	79
B.1. Crear un proyecto en <i>RStudio</i>	79
B.2. Funciones básicas en R	80
B.3. Estructuras de datos	81
B.4. Funciones	84

C. Asignaciones	87
C.1. Tarea 01: ¡Hola mundo con <i>Rmarkdown</i> !	87
C.2. Tarea 02: Ejercicios de crecimiento	89
C.3. Tarea 03: Colabora en el trabajo grupal	90
C.4. Tarea 04: Ejercicios con tablas de vida	91
C.5. Tarea 05: Matrices de transición	93
C.6. Tarea 06	94
C.7. Trabajo individual	95
C.8. Trabajo grupal	97

Capítulo 1

Introducción

La ecología de poblaciones se centra en el estudio de la dinámica de las poblaciones (su crecimiento e interacción con otras poblaciones), y en las interacciones de éstas con el ambiente. La ecología de poblaciones (también llamada **dinámica de poblaciones**) es un campo con un componente matemático y estadístico fuerte, y de gran importancia para la gestión de vida silvestre.

Algunas de las aplicaciones más importantes de esta disciplina, están relacionadas al cálculo de la viabilidad de poblaciones, al cálculo de tasas de extracción, e incluso a la creación de áreas protegidas dedicadas a proteger el ciclo de vida, o parte de éste, en determinadas especies.

El *Análisis de Viabilidad de Poblaciones*, es un ejemplo de una de las aplicaciones de la ecología de poblaciones para la gestión de vida silvestre. Este modelo predice el riesgo de que una población se extinga en una determinada cantidad de años. De esta manera, los gestores pueden modelar diferentes escenarios, cada cual con un conjunto específico de acciones de manejo, y decidir cuál de éstos es más efectivo en la conservación o manejo de la especie.

La creación de santuarios de pesca, por ejemplo, se fundamenta en el concepto de *Biogeografía de Islas* (REF) que también es parte de la ecología de poblaciones. Los santuarios de pesca funcionan como *fuentes*, es decir, zonas donde el crecimiento poblacional es positivo y existe migración de individuos. Éstos individuos, que se producen en exceso, migrarán hacia zonas de pesca, o extracción, para sostener actividades económicas. De esta manera, se garantiza la extracción sostenible en las zonas aledañas.

Algunos modelos importantes, como el modelo **bioeconómico**, que buscan la mayor rentabilidad económica por la extracción de una especie ([Grafton et al., 2006](#)), están basados en

modelos de crecimiento derivados de la dinámica de poblaciones. Este modelo estima la cantidad de esfuerzo extractivo que debe aplicarse a una especie, para mantener una rentabilidad positiva, y mantener un tamaño poblacional que garantice la continuidad de las poblaciones aprovechadas.

La dinámica de poblaciones es una de las ramas de la biología con un componente matemático y estadístico más fuertes. El desarrollo teórico de los modelos implica conocimiento de planteamiento y resolución de *ecuaciones diferenciales*. En la práctica, muchos problemas se plantean como ecuaciones diferenciales, pero no tienen solución analítica, por lo cual se requiere de conocimiento sobre *métodos numéricos*, *programación* o uso de lenguajes de programación. La mayoría de profesionales, no son desarrolladores teóricos, pero deben saber, al menos, sobre el uso de herramientas de análisis para esta disciplina.

Si el investigador conoce las herramientas de análisis, y quiere ponerlas en práctica, entonces requiere de conocimientos en *diseño experimental y muestral*; así como, *técnicas de muestreo* para conseguir los datos. Pero la limitación más fuerte, es el financiamiento requerido; ya que, la mayoría de los análisis tienen fuertes requerimientos de datos, y series de tiempo bastante amplias.

Capítulo 2

Modelos de crecimiento

HACER: conceptos de producción en exceso, de Darwin, y lucha por la existencia.

La evolución por selección natural implica que en una población que enfrente presiones para subsistir, existirán individuos mejor adaptados que otros. Algunos vivirán lo suficiente para reproducirse y otros no; además, dentro de aquellos que se reproduzcan, los más exitosos lo harán más frecuentemente, o con mayor descendencia. Este concepto implica que en una población debe haber suficiente variabilidad genética, que se refleje en un desempeño diferente en la reproducción, y que no todos los organismos vivirán lo suficiente para dejar descendencia o reemplazarse a sí mismos. Esto quiere decir, que las poblaciones deben de reproducirse y dejar un *exceso de descendencia*, para poder amortiguar el efecto sobre la reproducción de aquellos organismos que no logren reproducirse con éxito.

De esta manera, la sobre-producción de organismos es un requisito para que una población subsista en un intervalo prolongado de tiempo. Y la sobre-producción implica que las poblaciones tienen el potencial de *crecer*. La disciplina de la ecología de poblaciones, entonces, ha enfocado esfuerzos en modelar el crecimiento poblacional usando funciones matemáticas. Veremos las más básicas de ellas, con el objetivo de entender el origen y desarrollo de estos modelos.

El crecimiento en dinámica de poblaciones, está enfocado en la población, y no en el individuo. Algunos aspectos fisiológicos, e individuales, pueden ser importantes a la hora de modelar el crecimiento poblacional, y estos pueden ser incluidos como parámetros del modelo; pero, en general, el interés se centra en la estimación de la cantidad de individuos (o la biomasa) que conforma una población, y cómo cambia esta cantidad con respecto al tiempo.

El objetivo de los modelos de crecimiento, es obtener una función del tamaño de la po-

blación con respecto al tiempo. Existen dos aproximaciones principales para obtener esta función: la exponencial y la geométrica. El crecimiento exponencial se mide en cualquier momento en el tiempo, mientras que el crecimiento geométrico se mide a intervalos discretos. Es decir, ambos miden el crecimiento poblacional, pero una aproximación lo hace en intervalos continuos y la otra en intervalos discretos.

Las otra gran categoría de modelos de crecimiento tienen que ver con la dependencia de la densidad de población. Por ejemplo, una población con suficiente espacio y recursos, puede considerarse *denso-independiente*, mientras que una población que está en permanente competencia intraespecífica por la adquisición de espacio y recursos, tiene un crecimiento denso-dependiente.

2.1. Crecimiento denso-independiente

2.1.1. Crecimiento geométrico

Nuestra variable de interés es el tamaño poblacional, N . Queremos conocer el crecimiento poblacional desde año 0 ($t = 0$) hasta el año 1 ($t = 1$). Entonces, podemos restar $N_1 - N_0$ para encontrar dicho crecimiento, al que llamaremos ΔN ("Delta N"). De manera similar, podemos encontrar el crecimiento de la población en cualquier sub-intervalo de tiempo. Por ejemplo, si queremos conocer el crecimiento en el periodo $t = 1$ y $t = 0.5$, entonces nombramos este intervalo como Δt , y obtenemos el dato al dividir $\Delta N / \Delta t$. Esta razón corresponde a la *tasa de crecimiento*.

Una primer idea de cómo modelar la tasa de crecimiento, es pensar en que ésta equivale a la diferencia entre las *entradas* a la población (B , natalidad e inmigración) menos las *salidas* de la población (D , mortalidad y emigración):

$$\frac{\Delta N}{\Delta t} = B - D$$

Para conocer la tasa de crecimiento *per cápita*, dividimos la ecuación anterior por N :

$$\frac{\frac{\Delta N}{\Delta t}}{N} = \frac{B - D}{N}$$

Si la tasa de crecimiento per cápita es mayor a cero, entonces la población crece. Si es igual a cero, la población se mantiene estable. Si es menor a cero, la población decrece. Si asu-

mimos que la diferencia entra las entradas de la población y sus salidas son *constantes*, podemos arreglar la expresión anterior como $\frac{E-S}{N} = R_m$; con lo que obtenemos la forma familiar de la tasa de crecimiento:

$$\frac{\Delta N}{\Delta t} = R_m N \quad (2.1)$$

Sin embargo, la ecuación (2.1) aún no está en función del tiempo, que es el objetivo que se busca. Primero empecemos por predecir La población en el año uno (N_1) en función del tamaño de población inicial (N_0). Sabemos que N_1 será igual a N_0 más el crecimiento poblacional durante ese intervalo de tiempo. Es decir:

$$N_1 = N_0 + \frac{\Delta N}{\Delta t}$$

Y por la ecuación (2.1), substituyendo $N = N_0$, se tiene la relación:

$$\begin{aligned} N_1 &= N_0 + R_m N_0 \\ &= N_0 (1 + R_m) \\ &= N_0 \lambda \end{aligned} \quad (2.2)$$

Por tanto, el tamaño de población en el año uno, es igual al tamaño de población en el año cero, más el producto de la tasa de crecimiento per cápita por el tamaño de población en el año cero. Los arreglos posteriores, muestran que N_1 depende de N_0 y una constante $\lambda = 1 + R_m$, la cual representa la *tasa de multiplicación*. Entonces, la población crece cuando $\lambda > 1$, se mantiene estable si $\lambda = 1$, y decrece si $\lambda < 1$.

Ahora, podemos obtener N_2 al saber que $N_2 = N_1 \lambda$. Observamos que $N_1 = N_0 \lambda$; por tanto, sustituimos el valor de N_1 para acabar con $N_2 = N_0 \lambda \lambda = N_0 \lambda^2$. Si proseguimos de esta manera, concluimos que:

$$N_t = N_0 \lambda^t \quad (2.3)$$

Con lo que finalmente se logra el objetivo de tener una función del tamaño poblacional en relación al tiempo.

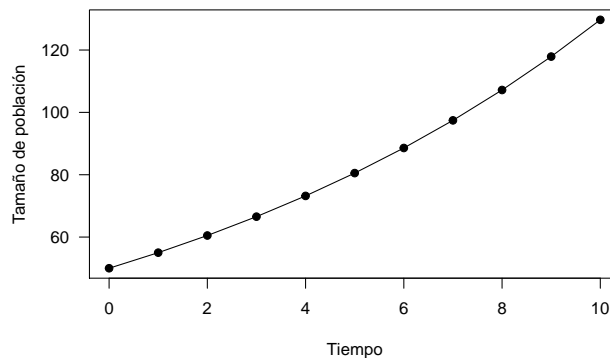
2.1.1.1. Ejemplos

Ejemplo 1 Graficar la ecuación (2.3)

Ahora que tenemos una relación del tamaño poblacional con el tiempo, podemos crear una función sencilla para observar su comportamiento.

```
plotGeomGrowth <- function(N0, lambda, t){
  vectorTiempo <- 0:t
  vectorPoblacion <- N0*lambda^vectorTiempo
  plot(vectorTiempo, vectorPoblacion,
        type = "p", xlab = "Tiempo", ylab = "Tamaño de población",
        las = 1, pch = 21, bg = 1)
  lines(vectorTiempo, vectorPoblacion)
}

plotGeomGrowth(50, 1.1, 10)
```



Ejemplo 2 ¿Cuál es el λ de una población que cuenta con 33 individuos en el año 0 ($t = 0$), y que tras 10 años cuenta con 25 individuos? Grafique la curva de crecimiento.

Al despejar la ecuación (2.3) para λ se tiene

$$\lambda = \left(\frac{N_t}{N_0} \right)^{\frac{1}{t}}$$

Substituyendo los valores correspondientes se tiene que $\lambda = 0.973$. Luego, usando la función creada en el ejemplo 1, y el recién calculado lambda, se grafica la curva de crecimiento.



2.1.1.2. Ejercicios

Ejercicio 1 Grafique la ecuación (2.1). En el eje y la tasa de crecimiento y en el eje x el tamaño poblacional. Utilice tres valores de R_m , uno positivo, uno igual a cero y otro negativo. El tamaño inicial de la población es de 50 individuos. $R_m \in [-1, 1]$, y $N \in [0, 50]$. Cuál es la representación gráfica de R_m en el gráfico.

Ejercicio 2 Grafique la ecuación (2.3). Utilice tres valores de λ , uno mayor a uno, otro igual a uno, y el último menor a 1, pero mayor a cero. El tamaño inicial de la población es de 50 individuos.

Ejercicio 3 **PICANTE** Todo libro de lógica matemática debe contener los métodos de demostración más comunes. Utilice el método de **inducción matemática** para demostrar que la ecuación (2.3) es válida para todo $n \in \mathbb{N}$ (números naturales). **5 % sobre la nota, dividido entre el número de estudiantes que respondan el ejercicio.**

Ejercicio 4 Si inoculo una población de bacterias en un medio de cultivo con suficiente espacio y nutrientes, con un estimado de 1×10^6 individuos, y tras tres horas, se estima una población de 3.5×10^6 individuos, ¿Qué valor tiene λ ? **NOTA.** En este caso, t representa una hora.

Ejercicio 5 Un cultivo de células dobla su tamaño poblacional en 15 minutos ($\lambda = 2$). Si se empieza con 1000 células, ¿cuántas de ellas existen tras 3 horas?

2.1.2. Crecimiento exponencial

En la sección anterior, se trabajó con intervalos de tiempo discretos. Pero si queremos conocer el tamaño poblacional en cualquier momento del tiempo, debemos trabajar con inter-

valos infinitamente pequeños. Esto quiere decir que la ecuación (2.3) se escribe en su forma continua:

$$\frac{dN}{dt} = r_m N \quad (2.4)$$

La ecuación (2.4) es una *ecuación diferencial de primer orden*¹. Este tipo particular de ecuaciones diferenciales tienen una solución analítica. Para este caso, se puede utilizar el método de separación de variables, para obtener la siguiente expresión del tamaño poblacional con respecto al tiempo (ver ejemplo 3):

$$N_t = N_0 e^{r_m t} \quad (2.5)$$

En la expresión anterior, r_m es la *tasa instantánea de crecimiento*, también conocida como la *tasa intrínseca de crecimiento natural*, o el parámetro de Malthus por Thomas Malthus. Este parámetro equivale a la diferencia entre la tasa intrínseca de nacimiento y la tasa intrínseca de mortalidad ($b - d$). La tasa intrínseca está relacionada con la tasa de multiplicación de la siguiente forma:

$$\lambda = e^{r_m}$$

$$r_m = \ln \lambda$$

El parámetro r_m tiene aplicaciones interesantes. Una de ellas es su facilidad para utilizarse en diferentes escalas de tiempo. Por ejemplo, si $r_m = 0.1$ por día, y queremos escalarlo a escala semanal, procedemos a multiplicar $r_m = 0.1 \times 7 = 0.7$. Al hacer esta transformación, se debe tener en cuenta la escala de tiempo con la que se interpretan y presentan los resultados.

2.1.2.1. Ejemplos

Ejemplo 3 Como obtener la ecuación de crecimiento exponencial (2.5) de la ecuación diferencial (2.4).

¹una ecuación diferencial existe cuando en la ecuación, la incógnita depende de su derivada. En este caso, si queremos despejar N , observamos que su derivada se encuentra en la expresión resultante

El método de separación de variables consiste en dejar todos los términos de la incógnita de un lado, y los términos de la variable independiente (t) del otro lado de la igualdad ([Barrantes Campos, 2015](#)). Entonces:

$$\frac{1}{N} \times \frac{dN}{dt} = r_m$$

Luego se integra ambos lados con respecto de la variable independiente:

$$\int \left(\frac{1}{N} \times \frac{dN}{dt} \right) dt = \int r_m dt$$

Observe que del lado izquierdo los diferenciales se cancelan:

$$\begin{aligned} \int \frac{dN}{N} &= r_m t + c \\ \ln N &= r_m t + c \end{aligned}$$

Se despeja N , y se obtiene $N = Ce^{r_m t}$. Luego, cuando $N = N_0$ entonces $t = 0$; por lo que la expresión se simplifica a $N_0 = Ce^0 = C$. Dando como resultado la expresión

$$N = N_0 e^{r_m t}$$

Ejemplo 4 De acuerdo con [Illman et al. \(2000\)](#) un gramo de *Chlorella emersonii* puede contener 29 kJ g^{-1} (energía por gramo). Si la tasa intrínseca es de 0.99 g d^{-1} , ¿cuántos gramos de *Chlorella* necesito para producir 5000 kJ ? ¿Cuál es el tiempo de producción? Asuma un crecimiento exponencial, y un inóculo inicial con $N_0 = 1 \mu\text{g}$ de *Chlorella*.

En este caso, pensamos en el tamaño poblacional como biomasa, en lugar del número de individuos. El primer paso es calcular N para producir la cantidad deseada de energía, lo cual resolvemos con una simple conversión para obtener:

$$N = \frac{1\text{g}}{29\text{kJ}} \times 5000\text{kJ} = 172.4138\text{g}$$

Luego, despejamos t de la ecuación (2.5):

$$t = \ln \left(\frac{N}{N_0} \right) r_m^{-1}$$

Se hacen las sustituciones correspondientes: $r_m = 0.99$, $N_0 = 1 \times 10^{-6}$ g, $N = 172.414$ g, y se obtiene que el tiempo necesario para obtener una biomasa equivalente a una energía de 5000 kilojoule es $t = 19.2$ d.

2.1.2.2. Ejercicios

Ejercicio 6 Si $\lambda = 1.027$ por semana. Escale λ de semanas a meses (1 mes = 4 semanas). Utilice la relación de $\lambda = e^{r_m}$.

Ejercicio 7 Para los siguientes escenarios de la ecuación (2.5):

- r_m negativo.
- r_m igual a cero.
- r_m positivo.

Obtenga el límite:

$$\lim_{t \rightarrow \infty} N(t)$$

Dé una interpretación de los resultados, en términos de la población.

Ejercicio 8 Demuestre, utilizando un razonamiento deductivo, que si $R_m < 0$, la población decrece. Puede usar los resultados del ejercicio 7.

Ejercicio 9 Analice biológicamente el significado del resultado del ejercicio 7, cuando r_m es positivo.

2.2. Crecimiento denso-dependiente

2.2.1. Crecimiento logístico

Nuestros dos modelos básicos funcionan en condiciones controladas, cuando el espacio y los recursos no son limitantes para el crecimiento. Sin embargo, esto no es lo que se observa en poblaciones silvestres, donde tras el periodo de crecimiento exponencial, sigue una disminución en la velocidad del crecimiento, hasta que llega a ser cero, o incluso negativo (decrecimiento).

Pueden existir varios mecanismos de regulación del crecimiento. Entre ellos existe la *competencia intraespecífica*, la *competencia interespecífica*, la *depredación*, entre otros. Algunas razones para la desaceleración del crecimiento pueden estar relacionadas a la tasa de consumo de alimento, frente a la tasa de producción. En este caso, cuando la tasa de consumo iguala a la tasa de producción, la población pierde el potencial de crecer. Los individuos empiezan a competir, puede haber emigración, mortalidad, etc.

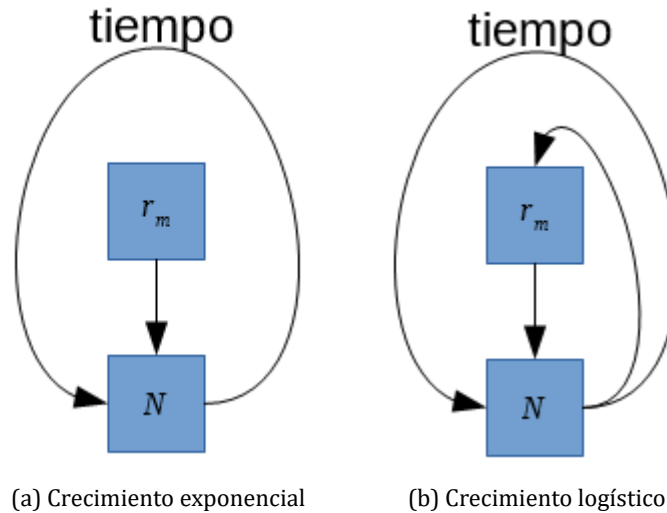


Figura 2.2.1: Diferencias en los modelos de crecimiento

Independientemente de las razones que causen la disminución del crecimiento, podemos entender que la tasa intrínseca de crecimiento (r_m), en realidad no es una constante como en el modelo de crecimiento exponencial (Figura 2.2.1a); más bien, es una función del tamaño de población, N (Figura 2.2.1b).

$$r_m = f(N_t)$$

Esta función debe tener algunas características particulares. Por ejemplo, debe ser máxima cuando el tamaño de población es pequeño, asumiendo que en ese momento hay muchos recursos y espacio para todos los individuos². Por otro lado, cuando la población alcanza un tamaño grande, la tasa de crecimiento debe disminuir hasta llegar a cero.

Debemos introducir un nuevo término a nuestro modelo de crecimiento, para cumplir con las características descritas arriba. Este término es la *capacidad de carga*, K , que es el punto donde la tasa de crecimiento se vuelve cero.

²Ignoramos el hecho de que en tamaños muy pequeños de población, existe problemas genéticos, o para encontrar pareja (Efecto Allee), que provocarían un crecimiento negativo.

Una expresión que cumple con estos requerimientos es:

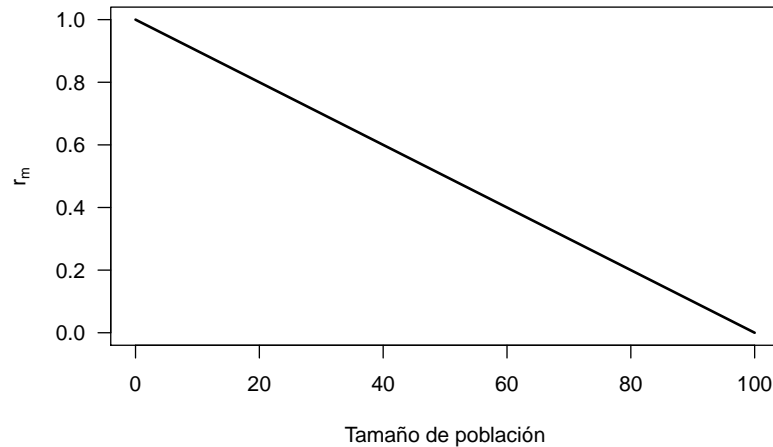
$$r_m = r_m \left(1 - \frac{N_t}{K} \right)$$

Podemos observar el comportamiento de r_m con un gráfico:

```
rm <- 1
K <- 100
N <- 0:100

val <- rm * (1 - (N / K))

plot(N,
      val,
      type = "l", las = 1, lwd = 2,
      xlab = "Tamaño de población",
      ylab = expression(r[m])
    )
```



Si ahora sustituimos la versión denso-dependiente de r_m en la ecuación (2.5), obtenemos

$$\frac{dN}{dt} = r_m N_t \left(1 - \frac{N_t}{K} \right) \quad (2.6)$$

Esta ecuación diferencial también puede resolverse por el método de separación de va-

riables (ver Ejemplo 3). Una vez resuelta, la expresión en función del tiempo es:

$$N_t = \frac{K}{1 + \left(\frac{K}{N_0} - 1\right) e^{-rt}} \quad (2.7)$$

Podemos graficar el comportamiento de la curva para una población hipotética con: $r_m = 0.4$, $K = 100$, y $N_0 = 10$. Para ello, usamos el siguiente código:

```
r <- 0.4
K <- 100
N0 <- 10
curve(
  K/(1+((K/N0-1)*exp(-r*x))),
  from = 0,to = 20,
  las = 1, lwd = 2,
  xlab = "Tiempo", ylab = "Tamaño de Población")
```

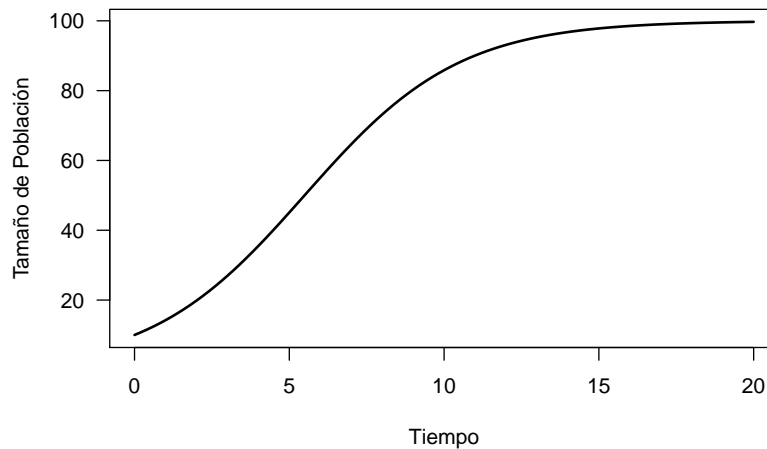


Figura 2.2.2: Crecimiento logístico, con capacidad de carga igual a cien individuos

Observamos que la población ya no crece de manera indefinida. Ahora tiene un tope superior igual a la capacidad de carga, K . Técnicamente, decimos que existe un límite asintótico al crecimiento; ya que, el tamaño de población *tiende* a K , pero nunca llega a alcanzarlo (pero ver ejercicio 10).

Por otro lado, observamos que la población crece rápidamente al inicio; pero, al final de la curva el crecimiento se detiene. Esto implica que en algún punto, el crecimiento alcanza un

máximo. Este punto se conoce como el *punto de inflexión*.

De cálculo diferencial, sabemos que los puntos de inflexión se obtienen al igualar la segunda derivada de la función a cero:

$$\frac{d^2N}{dt^2} = \frac{d(dN)}{d(dt)} = 0$$

Sabemos que la primer derivada de N_t corresponde a la ecuación (2.6). Entonces, para encontrar el punto de inflexión, primero debemos encontrar:

$$\frac{d^2N}{dt^2} = \frac{d(dN)}{d(dt)} = D \left[r_m N_t \left(1 - \frac{N_t}{K} \right) \right] = 0$$

Se utiliza la regla del producto, y se tiene:

$$D_{N_t} [r_m N_t] \left(1 - \frac{N_t}{K} \right) + r_m N_t D_{N_t} \left[\left(1 - \frac{N_t}{K} \right) \right] = 0$$

Al resolver las derivadas se obtiene:

$$N = \frac{K}{2}$$

Quiere decir, que el mayor crecimiento se obtiene cuando el tamaño de población es igual a la mitad de la capacidad de carga. Si sustituimos $N = K/2$ en la ecuación (2.6), tenemos que el crecimiento máximo de una población es³:

$$\left. \frac{dN}{dt} \right|_{N=K/2} = \frac{r_m K}{4}$$

Esto tiene grandes implicaciones en el manejo de recursos naturales. Es la base de los modelos de producción excedentaria. En pesquerías, a este número se le conoce como *Máximo Rendimiento Sostenible*; sin embargo, ha sido fuertemente criticado (Larkin, 1977), y en la actualidad se utilizan algunas variantes de esta cantidad, o otros modelos más apropiados, basados en estructura de edades o tallas.

³La línea vertical a mano izquierda en la ecuación indica que la derivada debe evaluarse en $N_t = K/2$.

2.2.2. Usos del modelo logístico

En la sección anterior calculamos el *Máximo Rendimiento Sostenible* (MRS). Ahora lo definiremos como:

El crecimiento máximo que una población puede producir, bajo una capacidad de carga determinada. Ésta es la cantidad máxima de individuos, o biomasa, que se puede extraer de una población, sin provocar un crecimiento negativo.

Recordando que:

$$\text{Tamaño de población} = \text{Tamaño anterior} + \text{Crecimiento}$$

La idea de *cosechar* una población se fundamenta en que si se extrae una cantidad igual al crecimiento de la población, la biomasa restante logrará regenerarse y crecer. La idea del MRS, es que el tamaño de una población cosechada, debe llevarse a $N/2$, para poder aprovechar el crecimiento máximo que puede generar dicha población.

Ejercicio 10 Demuestre que la población no crecerá más cuando llega a la capacidad de carga. Es decir, tome el límite de la ecuación (2.6), cuando $t \rightarrow \infty$.

Ejercicio 11 Suponga que existe un tanque sobre una balanza. Este tanque contiene aguas residuales, que son limpiadas por una pequeña planta del género *Lemna*. El flujo del tanque es tal, que la masa del agua siempre se mantiene constante; de modo que la balanza solo mide el crecimiento de *Lemna*. Nos interesa mantener una población de *Lemna* con un rápido crecimiento; ya que éste es proporcional a la tasa de extracción de toxinas del tanque. El tanque inicia con 1 kg de *Lemna*, con una tasa de crecimiento es de $5 \times 10^{-6} \text{ s}^{-1}$. Además, se ha determinado que el tanque solo soporta 100 kg de *Lemna*. ¿Cuál es el tamaño de población de *Lemna* que debería haber en el tanque para maximizar el crecimiento? ¿Cuánta biomasa debe extraer **en un día** para mantener un máximo de crecimiento? ¿A qué biomasa total debería cosechar la *Lemna*?

Ejercicio 12 De acuerdo con la definición del MRS, se asume que el crecimiento es igual a la cosecha. ¿Qué tan atinada es esta suposición?

Ejercicio 13 Grafique la tasa de crecimiento poblacional usando N_t como variable independiente. Asuma $K = 100$, $N_0 = 1$, $r_m = 1$.

2.2.3. Matrices

COMADRE COMADRE DATA Ojo a la guía de usuario

2.3. Otras fuentes bibliográficas

Esta sección está basada en los capítulos 4 y 5 de [Neal \(2004\)](#). En la sección 2.3 de [Berryman y Kindlmann \(2008\)](#), se desarrollan los mismos modelos básicos vistos aquí; en este mismo libro se presenta una buena introducción sobre la ecología de poblaciones como *sistemas*, en el capítulo 1.

Capítulo 3

Demografía

En el primer capítulo del material del curso, tratamos algunos modelos de crecimiento, donde factores importantes, como el sexo, la edad y el estado reproductivo de los organismos fueron ignorados. Anteriormente, consideramos a las poblaciones como una masa sin estructura que crece en relación al tiempo, y su propio tamaño. Esto tiene consecuencias importantes a la hora de contrastar las predicciones de los modelos, con datos reales sobre poblaciones.

Por ejemplo, en muchas poblaciones aprovechadas, los métodos de captura tienen alta selectividad para ciertas tallas o edades. Por tanto, existe un componente que provoca un efecto asimétrico en la mortalidad de una clase de edad a la siguiente.

Utilizar estructuras de edades en los modelos de dinámica de poblaciones, permite predecir algunos efectos de acciones de manejo. Por ejemplo, ¿qué pasa si distribuimos la mortalidad de manera equitativa en todas las clases de edad, o qué pasa si cargamos la mortalidad en cierta clase de edad pero no en las otras? Este tema se refleja en los reglamentos de tallas mínimas para la pesca en Costa Rica, por ejemplo.

La estructura por edades, puede entenderse como el aporte de cada clase de edad a la fecundidad, y a la supervivencia de una clase a la siguiente. Por ejemplo, podemos pensar en un modelo sencillo donde tenemos tres estados en una población: neonatos, juveniles y adultos. Los neonatos aportan individuos desde su clase, hacia los juveniles. Los juveniles pueden reproducirse tempranamente, y aportar individuos a los neonatos mediante la reproducción; además, aquellos que sobrevivan pasarán a la clase adulta. Por último, la clase adulta aportará muchos individuos a los neonatos mediante la reproducción, y algunos de ellos sobrevivirán y permanecerán en la misma clase en la siguiente generación (Figura [3.0.1](#)).

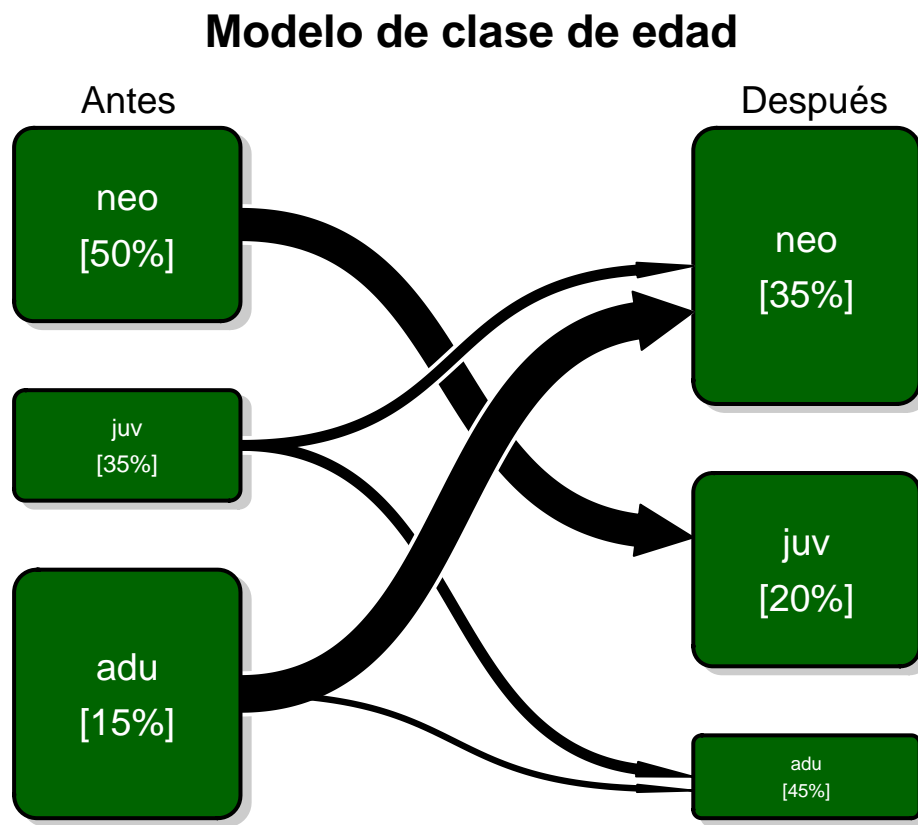


Figura 3.0.1: Ejemplo de las transiciones entre clases de edad

En las secciones posteriores, aprenderemos los fundamentos del uso de modelos estructurados por edad, y sus aplicaciones al manejo de recursos naturales. Para ayudar a la comprensión de la teoría, y su aplicación a datos reales, utilizaremos el paquete `popbio` de R.

Debemos instalarlo en la computadora, utilizando las siguientes líneas:

```
install.packages("popbio")
```

3.1. Tablas de vida

Las tablas de vida resumen un estudio longitudinal, que investiga el destino de los individuos de una *cohorte* durante su vida (o una muestra de la cohorte). Usualmente, la primera columna de una tabla de vida consiste del número de individuos en una clase de edad. Y las clases de edad, dependen del ciclo de vida de los organismos.

Este ordenamiento de los datos permite investigar cuántos individuos sobreviven de una clase de edad a la siguiente. Y a partir de aquí, se construyen otros valores poblacionales importantes.

Para construir una tabla de vida, debemos seguir los siguientes pasos¹:

1. x : Definir un intervalo de edad, que sea apropiado para el ciclo de vida de la especie. Puede indicarse únicamente el valor inicial del intervalo en la columna, pero lo ideal es trabajar con el punto medio del intervalo de clase. Por ejemplo, para el intervalo $[a, b)$, calculamos el punto medio del intervalo como $x = (a + b)/2$.
2. S_x : Indicar el número de supervivientes en esa clase de edad.
3. D_x : Anotar el número de muertes en esa clase de edad. Se calcula como $D_x = S_x - S_{x+1}$.
4. l_x : Calcular la proporción de sobrevivientes con respecto al número de individuos en la primera clase de edad S_0 . Para cada fila de la tabla de vida, calcular $l_x = S_x/S_0$.
5. d_x : Se calcula como D_x , pero la columna con los datos de entrada es l_x . Es decir: $d_x = l_x - l_{x+1}$.
6. q_x : La tasa de mortalidad, que se calcula como $q_x = D_x/S_x$ ó $q_x = d_x/l_x$.
7. e_x : La esperanza de vida promedio. Esta se calcula al sumar todos los sobrevivientes desde la clase de edad x hasta la última, dividido entre el número de supervivientes en

¹en poblaciones silvestres de animales, generalmente se trabaja solo con las hembras

la clase x . Esta manera de calcular e_x supone que todos los individuos sobreviven desde el inicio de la clase de edad hasta el final y luego mueren. Para corregir restamos a este valor la mitad de la amplitud del intervalo en la clase de edad x . Es decir, si la amplitud del intervalo es un año, entonces restamos $0.5 \|x\| = 0.5(1)$. La fórmula Si tenemos n clases de edad, entonces calculamos $e_x = (\sum_{i=x}^n S_i) / S_x - 0.5 \|x\|$.

Los cálculos anteriores son interesantes, pero aún hay datos que se pueden añadir para sacar todo el provecho de las tablas de vida: los datos de reproducción. En el caso de animales, las tablas suelen trabajarse con el número de hembras; además, pueden añadirse datos sobre la descendencia que produce cada hembra (contando también solo los individuos de este sexo). Por lo tanto, a las columnas anteriores añadimos las siguientes:

1. m_x : El número de descendientes hembras por cada hembra de la clase x
2. $m_x l_x$: La producción de nuevas hembras por clase de edad.
3. $x m_x l_x$: paso intermedio para calcular parámetros reproductivos.

Y con los resultados de una tabla como esta, podemos calcular parámetros importantes como la tasa de crecimiento neta (R_0), que es la cantidad promedio de descendientes hembras que produce una hembra a lo largo de su vida. También podemos calcular el tiempo medio de generación (T_c), que es tiempo que en promedio dura una hembra en producir una cantidad de descendientes hembra igual a la tasa de crecimiento neta, R_0 . Estos parámetros los calculamos como:

$$R_0 = \sum l_x m_x$$

$$T_c = \sum x m_x l_x$$

El R_0 es un reflejo de la tasa de multiplicación (λ , ver sección 2.1.1). Es decir, la población crece si $R_0 > 1$, se mantiene estable si $R_0 = 1$, y decrece si $R_0 < 1$.

También podemos relacionarlo con la tasa intrínseca de crecimiento per cápita r_m (ver sección 2.1.2). Para ello usamos la siguiente relación:

$$r_m \approx \frac{\ln(R_0)}{T_c}$$

3.1.1. Tablas de vida sobre cohortes (tablas horizontales)

Este tipo de tablas se construyen al seguir un grupo de organismos de una misma cohorte o generación. Es más utilizado en plantas o animales sésiles; sin embargo, con el financiamiento y el equipo correcto, puede ser utilizado en animales móviles.

El paquete **popbio** de R, ofrece varios conjuntos de datos. Uno de ellos sobre una planta llamada *Aquilegia*. El conjunto de datos contiene información sobre cada planta, y su destino después de muestreos anuales.

```
library(popbio) # cargar paquete
data("aq.trans") # cargar datos
aq.trans -> dat; rm(aq.trans) # crear objeto de trabajo

# arreglar niveles del factor
dat$status <- factor(dat$stage, levels(factor(dat$stage))[1:4])
attach(dat)
head(dat, 5) # muestra las primeras 5 filas de datos
```

```
##   plot year plant  stage leaf rose fruits  fate rose2 status
## 1  903 1996     1  small    0   0      0 small   NA  small
## 2  903 1996     2 flower   NA  NA      1 large   NA flower
## 3  903 1996     3  small    0   0      0 large   NA  small
## 4  903 1996     4  small    1  NA      0 small    0  small
## 5  903 1996     7  small    2  NA      0 large   NA  small
```

```
# Construimos una matriz con los datos de sobrevivencia
(pop <- table(year, status))
```

```
##           status
## year  recruit small large flower
## 1996      12   134   17    62
## 1997     287    75   68     6
## 1998     186    60   41    80
## 1999      76    84   57    74
## 2000       5    58   59    52
## 2001       5    31   58     8
## 2002       0    13   16     0
## 2003       3     6    4     0
```

```
#Escogemos una diagonal de la matriz (cohorte)
Sx <- numeric(4)
for(i in 1:4){
  Sx[i] <- pop[(i+2),i]
}

aqData <- tablaVida(x.sup = 1:4, Sx = Sx)
rm(dat, pop, i) # Limpiar un poco
```

Si aplicamos las fórmulas que hemos visto en clase, obtenemos la siguiente tabla de vida:

Cuadro 3.1.1: Tabla de vida para una cohorte de *Aquilegia*.

x	S_x	D_x	l_x	d_x	q_x	e_x
1	186	102	1.000	0.548	0.548	1.312
2	84	25	0.452	0.134	0.298	1.298
3	59	51	0.317	0.274	0.864	0.636
4	8	NA	0.043	NA	NA	0.500

3.1.2. Tablas de vida estáticas

Para algunos animales grandes y longevos, no es práctico construir tablas de vida horizontales. Así que se toman datos en un momento determinado de tiempo, y la tabla se construye a partir de esta información.

Tablas de vida verticales o tiempo-específicas— Estas tablas se construyen cuando se conoce la edad de los organismos. Entonces, se cuentan cuántos individuos hay en cada clase de edad; posteriormente, se cuentan las muertes en cada clase de edad, y se combina la información para construir una tabla de vida.

Tablas estacionarias basadas en la edad a la muerte— Estas tablas se construyen con información de cadáveres, o partes de éstos. La edad en los cadáveres debe poder ser establecida claramente para que el método funcione. En [Neal \(2004\)](#), se explica el método con un ejemplo basado en el carnero de Dali *Ovis dalli*. Donde el investigador recolectó cráneos y determinó la edad de cada uno.

Por ejemplo, si recolecto 1000 mandíbulas de algún mamífero, y puedo datar estos huesos, el primer paso es asumir que estos 1000 individuos constituyen la primer clase de edad.

Luego, si el número de mandíbulas en la segunda clase de edad S_2 es 600, resto el S_x anterior a esta cantidad, y así calculo D_1 . Es decir, $D_1 = 1000 - 600$. Y así sucesivamente.

Tablas estacionarias basadas en la estructura de edad de la población— Si la población no cambia en tamaño (la tasa de natalidad iguala la tasa de mortalidad), y si la estructura de edad es un reflejo de la Supervivencia (S_x): podemos asumir que la edad de los individuos puede ser datada de manera precisa, y que hemos obtenido una muestra representativa de la población (Neal, 2004). Esto quiere decir, que podemos utilizar la frecuencia en cada clase de edad, y asumir que corresponde a la supervivencia. Y que el cambio de frecuencia de una clase de edad a la siguiente se debe a la mortalidad (D_x).

3.2. Curvas de supervivencia

Las tablas de vida pueden mostrar algunos detalles de las estrategias de supervivencia de los organismos. Por ejemplo, ¿tienen cuidado parental, o confían en un gran número de juveniles (o huevos, larvas, semillas) de los cuales un pequeño porcentaje de ellos llegarán a ser adultos?

Para ello, podemos utilizar directamente la columna S_x y graficarla contra la columna x . Sin embargo, si queremos comparar entre distintas especies, o contra diferentes experimentos o estudios, lo ideal es utilizar una columna normalizada como l_x . En algunos casos, para mejorar la visualización de los datos, se multiplica la columna l_x por un factor, generalmente un factor de mil, para mostrar la supervivencia por cada mil individuos.

También se puede explorar la mortalidad en cada clase de edad, siguiendo un procedimiento similar; pero, reemplazando l_x por q_x . Para ilustrar el procedimiento podemos utilizar los datos del Cuadro 3.1.1:

```
op <- par()
par( mar = c(5, 5, 4, 5) + 0.1)

plot(aqData[,1],aqData[,4]*1000,
      xlab = "Edad", ylab = "", axes = FALSE, type = "n")

axis(side = 2, at = c(0,500,1000))
lines(aqData[,1],aqData[,4]*1000, lwd = 2)
mtext(side = 2, "Supervivencia x 1000", line = 3)
```

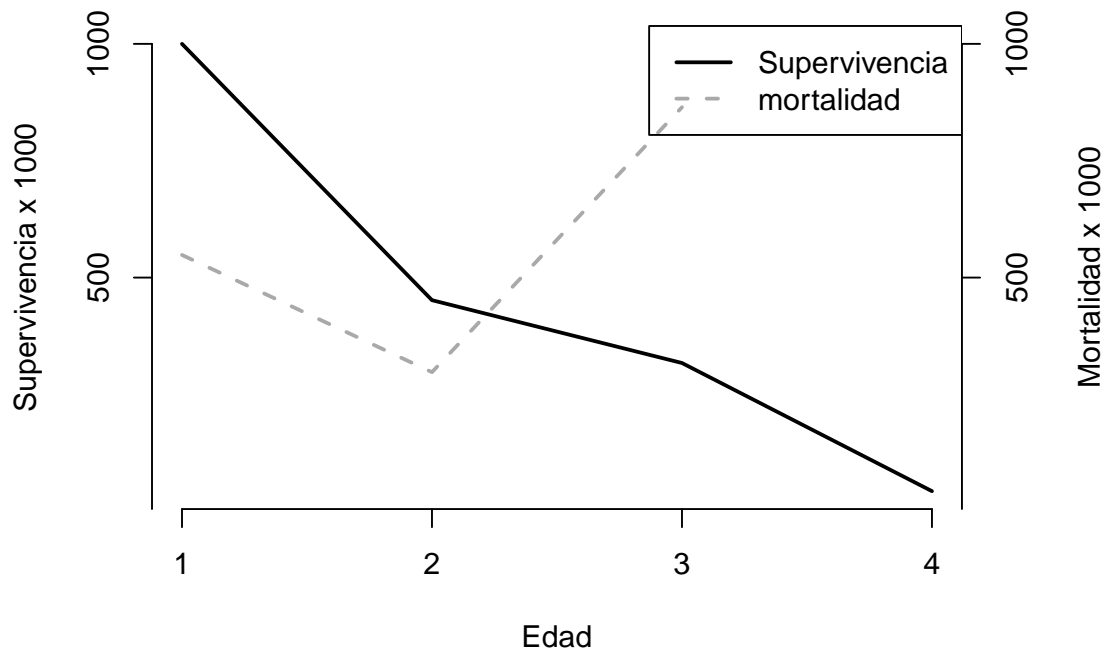


Figura 3.2.1: Supervivencia y mortalidad de *Aquilegia*

```
axis(side = 1, at = 1:4)

axis(side = 4, at = c(0,500,1000))
lines(aqData[,1],aqData[,6]*1000, lwd = 2, lty = 2, col = "darkgray")
mtext(side = 4, "Mortalidad x 1000", line = 3)

legend("topright", legend = c("Supervivencia", "mortalidad"),
      lty = c(1,2), lwd = c(2,2), col = c(1, "darkgray"))

par(op)
rm(aqData)
```

3.3. Análisis de Curvas de Captura

Las curvas de captura fueron desarrolladas para analizar datos en pesquerías. Su objetivo es determinar la tasa de supervivencia, o mortalidad, para una población. Este método asume que todas las clases de edad tienen una misma tasa de supervivencia.

Es una herramienta general que puede ser utilizada con tablas de vida estacionarias. En general, el análisis de curvas de captura puede ser utilizado en casos donde se tengan frecuencias de individuos por clase de edad.

El análisis de curvas de captura, también puede utilizarse con datos de tallas; siempre y cuando, exista una función que relacione las tallas con la edad. En este último caso, hay que considerar la incertidumbre que existe al convertir tallas a edades ².

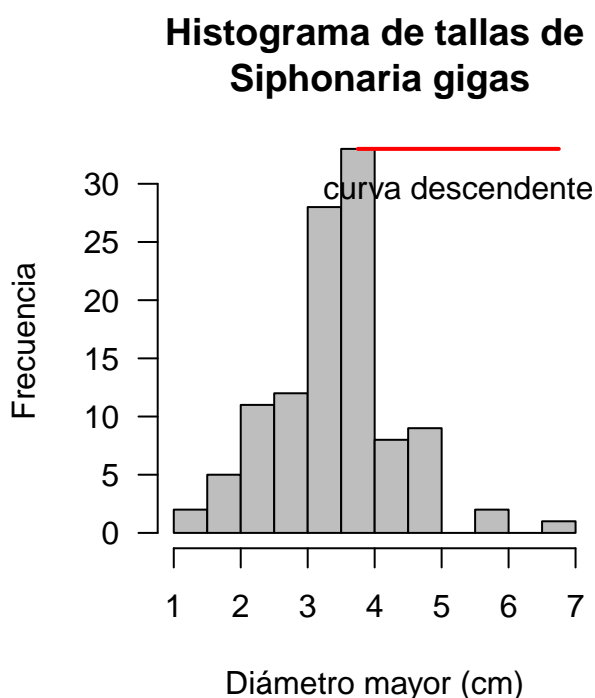


Figura 3.3.1: Histograma de tallas de *Siphonaria gigas* en el RNVS Playa Hermosa-Punta Mala. Datos de 2017-2018.

En la mayoría de los casos, tanto con datos de edades como tallas, existirán clases pobremente representadas (Figura 3.3.1). Esto se debe a que la probabilidad de captura con ese método específico varía en función de la edad o talla. También puede significar que existe una

²Una manera de hacerlo es utilizar simulaciones de Monte Carlo.

separación espacial en las diferentes edades de la especie; y por tanto, que no fueron representadas en el muestreo.

Cuando esto sucede, se utiliza la sección descendente de la curva, y se asume que todas las clases contenidas en esa sección están bien representadas. Una vez que se ha seleccionado el sub-conjunto de datos con el cual se hará la estimación, se pueden utilizar varios métodos para estimar la tasa de supervivencia (o mortalidad).

3.3.1. Método de Chapman y Robson 1960. Estimación de supervivencia usando todas las clases de edad

Este método se deriva de asumir que los conteos de individuos supervivientes provienen de una distribución de probabilidad multinomial³. En cada una de las clases de edad, la probabilidad de obtener un individuo de edad x tiene una distribución geométrica: $f_x(x) = (1 - s)s^x$. Donde s es la probabilidad de sobrevivencia que se desea estimar.

El desarrollo del estimador de s , se detalla en [Skalski et al. \(2005\)](#). Y se basa en el concepto de *máxima verosimilitud*. Su fórmula es sencilla:

$$\hat{s} = \frac{T}{n + T - 1}$$

Donde n es el tamaño de la muestra. $T = \sum_{i=1}^n x_i$, es decir la suma de las edades de todos los individuos en la muestra. La varianza de \hat{s} , se calcula como:

$$\text{Var}(\hat{s}) = \hat{s} \left(\hat{s} - \frac{T - 1}{n + T - 2} \right)$$

Suponer que todas las clases de edad mantienen una tasa de supervivencia común es arriesgado. Las suposiciones que se deben cumplir para aplicar este modelo con seguridad son:

1. Existe una estructura de edad estable.
2. La población es estacionaria.
3. Todos los animales tienen una misma probabilidad de selección(captura).
4. La muestra es representativa de la población.

³La distribución multinomial es una versión de la versión binomial para más de dos estados posibles de la variable.

5. El destino de todos los animales es independiente.
6. Todas las edades se estiman sin error.
7. La probabilidad de supervivencia anual es constante a través de todas las clases de edad.

3.3.2. Método de regresión

De manera alternativa, se puede utilizar una simple regresión lineal para estimar una tasa de supervivencia común a todas las clases de edad. Para ello, asumimos que el número esperado de individuos contados en una clase de edad x (S_x), es igual un número de reclutas N_0 , que se mantiene constante a través del tiempo; multiplicado por la probabilidad de que un individuo sea capturado en la muestra p ; multiplicado por probabilidad de haber sobrevivido hasta la clase x (s^x). Es decir:

$$E(S_x) = N_0 p s^x$$

Si tomamos logaritmo a ambos lados de la expresión anterior, obtenemos:

$$\ln(\bar{S}_x) = \ln(N_0 p) + x \ln(s)$$

Que presenta la forma familiar de una línea recta: $y = a + bx$. Lo importante aquí, es reconocer que $s = e^b$. Así que al estimar los parámetros de la recta, podemos derivar la tasa de supervivencia s .

Ejemplo 5 *Calcular la tasa de supervivencia, usando los datos de Aquilegia (Cuadro 3.1.1)*

```
#Definir los datos para la regresión
y <- log(Sx) #Sx, es el conteo de supervivientes de Aquilegia
x <- 1:4 # Las clases de edad

salida <- glm(formula = y~x, family = gaussian(link = "identity"))
s <- signif(
  exp(coef(salida)[2]),
  digits = 3)

sLim <- signif(
```

```
exp(confint(salida)[2,]),
digits = 3)
```

Con el código de arriba, obtenemos que $s = 0.376$, con un intervalo de confianza al 95 % de 0.234–0.602.

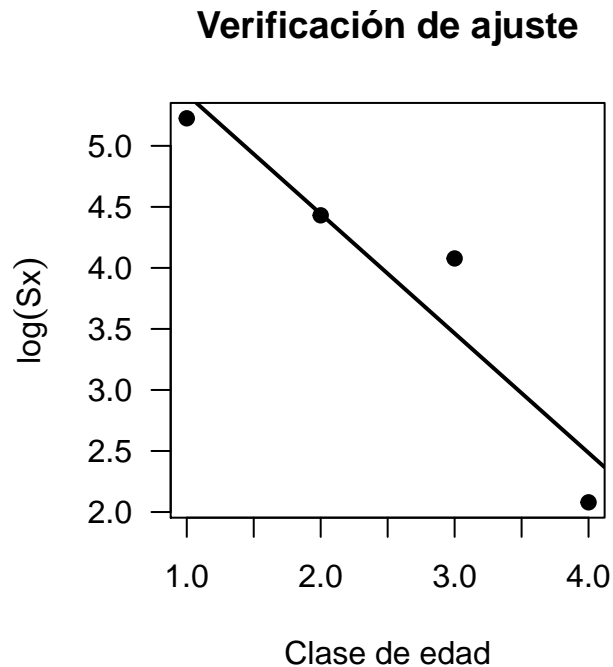


Figura 3.3.2: Verificación de ajuste del modelo a los datos de extitAquilegia. Los puntos debería aparecer alineados con la recta de mejor ajuste

3.3.3. Ejercicios

Ejercicio 14 ¿Cual es el significado ecológico de R_0 ?

Ejercicio 15 Si a usted le piden hacerse cargo de determinar el éxito de un programa de reintroducción, basado en el éxito reproductivo de la especie: Enumere los pasos a seguir para determinar si la especie se reproduce exitosamente, utilizando las herramientas vistas en las tablas de vida? Asuma que ud puede muestrear una cohorte de principio a fin.

Ejercicio 16 Calcule si la siguiente población está creciendo, se mantiene estable o está decreciendo:


```
##          x  Sx  mx
## [1,]    5  20  0.0
## [2,]   10  45  0.0
## [3,]   15  10  1.2
## [4,]   20   5  1.1
```

Ejercicio 17 En [Neal \(2004\)](#) se presentan los datos sobre el estudio de *Ovis dalli*. Calcule la tabla de vida:

```
##          x  Sx
## [1,]    1 608
## [2,]    2 487
## [3,]    3 480
## [4,]    4 472
## [5,]    5 465
## [6,]    6 447
## [7,]    7 419
## [8,]    8 390
## [9,]    9 348
## [10,]  10 268
## [11,]  11 154
## [12,]  12  59
## [13,]  13   4
## [14,]  14   2
## [15,]  15   0
```

Ejercicio 18 Grafique la curva de supervivencia y mortalidad del ejercicio anterior.

Ejercicio 19 Estime una tasa de supervivencia común a todas las clases de edad de *Ovis dalli*, usando los datos del ejemplo 17. Use el método de análisis de curvas de captura por regresión.

3.4. Matrices de transición

En la sección anterior, nos enfocamos en el cálculo de algunos parámetros de crecimiento poblacional utilizando tablas de vida. Las tablas de vida son útiles cuando tenemos clases de edad bien definidas, sin embargo, en algunos casos es más conveniente estructurar una población por *etapas*, y no por clases de edad. Para este caso, el uso de matrices es más apropiado.

Modelo poblacional de la ballena asesina

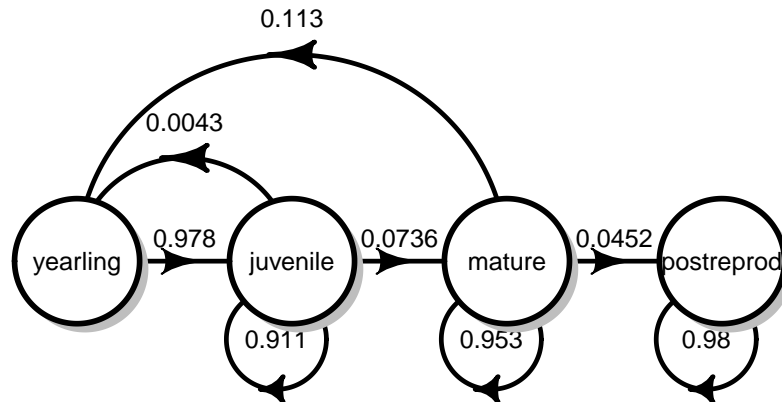


Figura 3.4.1: Un diagrama que representa gráficamente una matriz de transición

Por ejemplo, algunos insectos pueden ser fácilmente descritos por etapas: huevo, larva, pupa y adulto. Igual que algunas plantas: semilla, plántula, flor, fruto. Incluso, los modelos matriciales son útiles en animales cuyas etapas de vida están mejor diferenciadas por el comportamiento, más que por la edad; es decir, la supervivencia en cada etapa está más relacionada con el comportamiento, que con la edad del individuo.

El paquete **popbio** de R, contiene un conjunto de datos sobre la ballena asesina. En este conjunto de datos, la estructura de la población está mejor representada por sus etapas (Figura 3.4.1). Entendemos del diagrama, que las flechas nacen de la etapa inicial y apuntan hacia la etapa a la cual se hará la transición. Vemos que en algunos casos, las flechas salen desde una etapa, y apuntan a sí mismas; en este caso, se habla de permanencia (P). Si la etapa apunta a otra de mayor edad, se habla de crecimiento (G). Finalmente, si la etapa apunta a una de menor edad se habla de fecundidad (F).

Las **matrices de transición** contienen la información sobre la probabilidad de pasar de un estado a otro (como en la figura 3.0.1 también). Por ejemplo, para el diagrama de la ballena asesina, la matriz asociada es:

Cuadro 3.4.1: Matriz de transición para la ballena asesina.

		Desde etapa			
		yearling	juvenile	mature	postreprod
Hasta etapa	yearling	0.0000	0.0043	0.1132	0.0000
	juvenile	0.9775	0.9111	0.0000	0.0000
	mature	0.0000	0.0736	0.9534	0.0000
	postreprod	0.0000	0.0000	0.0452	0.9804

Por convención, la matriz se lee desde las columnas hacia las filas. Es decir, si queremos encontrar la probabilidad de pasar de la etapa adulta, a la post-reproductiva, primero se ubica la columna de los adultos (*mature*), y luego se busca la fila de los post-reproductivos (*postrep*), y se lee la celda que interseca esa columna con esa fila. Para este caso es 0.0452.

3.4.1. Proyección de una matriz a N_{t+1}

Uno de los usos de las matrices de transición (también llamadas matrices poblacionales), es la proyección del tamaño de población en el tiempo. Esto es lo que el modelo de *Crecimiento Geométrico* hacía, pero con un tamaño de población global, sin estructura por etapas o edades.

Para proyectar una matriz, necesitamos entender la **multiplicación matricial**. Esta es una operación básica que se enseña en los cursos de álgebra lineal, y consiste en multiplicar los elementos de una matriz con los elementos de otra matriz o vector. Pero para ello, hay un orden específico.

Retomando la matriz de transición de la ballena asesina, pero sustituyendo los valores numéricos por símbolos, tenemos:

Cuadro 3.4.2: Matriz de transición para la ballena asesina, expresada de manera simbólica.

	yearling	juvenile	mature	postreprod
yearling		F_{jy}	F_{my}	
juvenile	G_{yj}	P_j		
mature		G_{jm}	P_m	
postreprod			G_{mP}	P_P

Vamos a llamar a esta matriz **A**. Luego, para la proyección es necesario contar con un vector que contenga el número de individuos en el año (o unidad correspondiente) previo al

que se desea proyectar. Es decir, si queremos proyectar a $t + 1$, debemos conocer el tamaño de población en el año t .

$$\mathbf{n}_t = \begin{pmatrix} n_y \\ n_j \\ n_m \\ n_p \end{pmatrix}$$

Luego, representamos la operación como:

$$\mathbf{n}_{t+1} = \mathbf{A}\mathbf{n}_t$$

La expresión anterior es una forma corta de representar la multiplicación de matrices. Tanto los vectores como las matrices pueden ser representados en negrita; las matrices en mayúscula; los vectores en minúscula. Esta expresión simplifica la siguiente operación:

$$\mathbf{n}_{t+1} = \begin{pmatrix} 0 & F_{jy} & F_{my} & 0 \\ G_{yj} & P_j & 0 & 0 \\ 0 & G_{jm} & P_m & 0 \\ 0 & 0 & G_{mp} & P_p \end{pmatrix} \times \begin{pmatrix} n_y \\ n_j \\ n_m \\ n_p \end{pmatrix}$$

El resultado de esta multiplicación es un arreglo con el número de filas iguales al primer multiplicando (la matriz), y el número de columnas iguales al segundo multiplicando (el vector). Por tanto, el resultado es un nuevo vector.

El i -ésimo elemento del vector \mathbf{n}_{t+1} , se calcula como: $\sum_{j=1}^k \mathbf{A}_{i,j} \cdot \mathbf{n}_j$. Todo el proceso se resume a continuación:

$$\mathbf{n}_{t+1} = \begin{pmatrix} 0 \cdot n_y + F_{jy} \cdot n_j + F_{my} \cdot n_m + 0 \cdot n_p \\ G_{yj} \cdot n_y + P_j \cdot n_j + 0 \cdot n_m + 0 \cdot n_p \\ 0 \cdot n_y + G_{jm} \cdot n_j + P_m \cdot n_m + 0 \cdot n_p \\ 0 \cdot n_y + 0 \cdot n_j + G_{mp} \cdot n_m + P_p \cdot n_p \end{pmatrix} \quad (3.1)$$

Ejemplo 6 *Hacer una proyección a $t + 1$ con la matriz de transición de la ballena asesina*

Vamos a asumir que se realizó un censo poblacional, y que los datos de abundancia se clasificaron según las etapas de la matriz de transición (Cuadro 3.4.1). Los resultados del censo fueron:

$$\mathbf{n}_t = \begin{pmatrix} 123 \\ 87 \\ 60 \\ 64 \end{pmatrix}$$

Si sustituimos los valores de la matriz de transición, y los datos del censo en la ecuación (3.1), tenemos:

$$\mathbf{n}_{t+1} = \begin{pmatrix} 0 \cdot 123 + 0.0043 \cdot 87 + 0.1132 \cdot 60 + 0 \cdot 64 \\ 0.9775 \cdot 123 + 0.9111 \cdot 87 + 0 \cdot 60 + 0 \cdot 64 \\ 0 \cdot 123 + 0.0736 \cdot 87 + 0.9534 \cdot 60 + 0 \cdot 64 \\ 0 \cdot 123 + 0 \cdot 87 + 0.0452 \cdot 60 + 0.9804 \cdot 64 \end{pmatrix}$$

El resultado es:

$$\mathbf{n}_{t+1} = \begin{pmatrix} 7.17 \\ 199.5 \\ 63.6 \\ 65.5 \end{pmatrix}$$

En R:

```
library(popbio)
data(whale)

nt <- c(123, 87, 60, 64)

ntp1 <- whale%*%nt # el '%*%' representa el operador para multiplicar
                  # matrices en R

ntp1

##           [,1]
## yearling    7.17
## juvenile  199.50
## mature     63.61
## postreprod 65.46
```

```
#limpiar  
detach("package:popbio", unload = TRUE)  
rm(whale, ntp1, nt)
```

3.4.2. El paquete popbio

En la sección anterior se explicó cómo usar una matriz de transición para proyectar una población de t a $t + 1$. Este proceso puede hacerse de manera iterativa a través de un ámbito de tiempo determinado. El resultado de esta operación es una Distribución Estable de Edad (*SAD* por sus siglas en inglés) (Skalski *et al.*, 2005). Si la población mantiene una fecundidad específica por edad y tasas de supervivencia estables, ésta alcanzará una SAD que no depende de la población inicial. Esta condición es útil para calcular parámetros poblacionales la tasa de multiplicación λ , la tasa reproductiva neta R_0 , el tiempo de generación T_c , etc., aunque los detalles del cálculo quedan fuera del alcance del curso. En lugar de ello, se escoge una aproximación más práctica para calcular λ usando el paquete **popbio** (Stubben y Milligan, 2007). A continuación se presenta un resumen de las principales funciones del paquete **popbio**.

- `pop.projection(TM, n0, t)`: Proyecta una población. *TM*: matriz de transición; *n0*: vector de tamaño de población inicial; *t*: cantidad de tiempo que se desea proyectar. Devuelve valores como `$lambda`: λ , `$stable.stage`: la proporción estable de clases de edad, o etapas, en una población; `$stage.vectors`: la cantidad de individuos en cada clase de edad o etapa, desde el tiempo cero hasta el tiempo *t*; `$pop.sizes`: vector con el tamaño de la población total, desde el tiempo cero hasta el tiempo *t*; `$pop.changes`: el λ desde el tiempo cero hasta el tiempo *t*.
- `stage.vector.plot(stage.vectors, col)`: grafica la matriz de transición con respecto al tiempo. `stage.vectors`: recibe la salida de la función `pop.projection`, específicamente el vector: `$stage.vectors`. El argumento `col` es una secuencia de enteros, con las columnas que se desean graficar.
- `fundamental.matrix(TM)`: Recibe una matriz de transición y devuelve una matriz fundamental, con los errores de estimación. La matriz fundamental tiene la misma estructura que una matriz de transición, pero sus elementos contienen el número de unidades de tiempo que pasa cada individuo en una *etapa* de crecimiento. Es apropiada para poblaciones con crecimiento estructurado por etapas.
- `net.reproductive.rate(TM)`: Devuelve el R_0 de una matriz de transición.

- `generation.time(TM)`: Devuelve el T_c de una matriz de transición.
- `stoch.projection(matrices, n0, tmax, nreps, ...)`: Genera una proyección estocástica de una población. `matrices`: Lista de matrices de transición (creadas a partir de la estimación de una matriz con valores de tendencia central y de dispersión. `n0`: vector con tamaño inicial de población; `tmax`: entero con el número de pasos de tiempo que quieren predecirse a futuro; `nreps`: número de pasos a iterar. Los detalles del método se encuentran en [Elder et al. \(2003\)](#).
- `stoch.growth.rate(matrices)`: Versión estocástica para calcular *el logaritmo* de la tasa de crecimiento.
- `stoch.quasi.ext(matrices, n0, Nx, tmax, maxruns, nreps)`. Hace una proyección de la probabilidad de extinción de una población. Cuyos argumentos son:
 - `matrices`: Una lista con dos o más matrices de transición
 - `n0`: Vector con tamaño de población inicial.
 - `Nx`: Umbral de quasi-extinción.
 - `tmax`: número de pasos a proyectar en el futuro.
 - `maxruns`: Número de veces que se simula la distribución acumulativa sobre la probabilidad de extinción.
 - `nreps`: Número de iteraciones.

3.4.2.1. Estimación de parámetros básicos con

Para simplificar las cosas, las funciones del paquete **popbio** se han reunido en dos *clases de referencia*. Las clases de referencia son objetos en R que contienen los datos y los métodos para analizarlos en un solo objeto. Las funciones para crear estas clases de referencia se encuentran en el archivo `Source.R` dentro del repositorio de este material.

La clase `transMat` toma como argumentos `a: matriz`: una matriz de transición; `n0`: un vector con el tamaño de población inicial; `t`: el número de pasos de tiempo a proyectar la población. Por ejemplo, para calcular los parámetros de crecimiento de la ballena asesina, asumiendo una población inicial de $n_0 = \{100, 80, 40, 20\}$ utilizamos el siguiente comando:

```
library(popbio)
data(whale)
```

```
whaleTM <- transMat$new(matriz=whale,n0 = c(100,80,60,40),t = 15)
```

Al crear este objeto, se ha utilizado un *constructor* con el nombre de la clase `transMat` seguido de la instrucción `$new(...)`, que indica a R que se quiere crear una nueva instancia de esta clase. Seguidamente le pasamos los datos a los argumentos `matriz=whale`, `n0 = c(100,80,60,40)`, `t = 15`. Y guardamos todo en un nuevo objeto llamado `whaleTM`. Este objeto contiene los datos, pero también las funciones para calcular los parámetros poblacionales⁴

Para graficar los resultados de la proyección utilizamos el comando `whaleTM$plotTM()` (Figura 3.4.2).

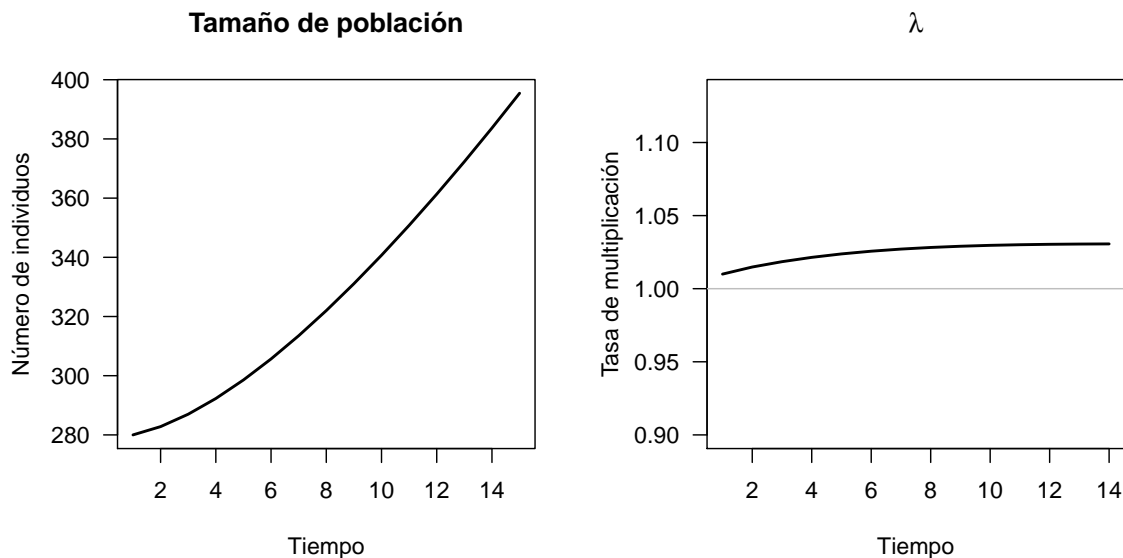


Figura 3.4.2: Resultados de la proyección de la matriz de transición

Para graficar los resultados la proporción de la población en cada etapa existe el comando `whaleTM$plotStage()` (Figura 3.4.3).

Las demás funciones para calcular parámetros poblacionales se presentan abajo:

```
# La tasa de multiplicación
whaleTM$darLambda()
```

```
## [1] 1.03
```

⁴Estas funciones son una llamada a las funciones del paquete **popbio**. Su objetivo es simplificar la lectura del código y la resolución de los ejercicios.

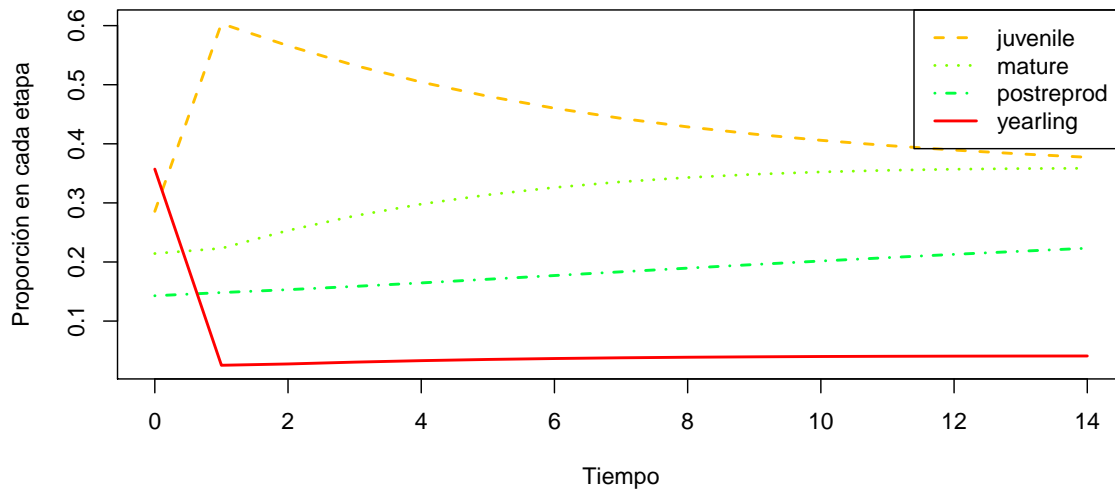


Figura 3.4.3: Proporción de la población en cada etapa de edad

```
#La tasa de crecimiento neto
```

```
whaleTM$darR0()
```

```
## [1] 2.01
```

```
#El tiempo de generación
```

```
whaleTM$darTc()
```

```
## [1] 27.9
```

3.4.3. Viabilidad de poblaciones

Los gestores de vida silvestre utilizan las proyecciones de población para evaluar escenarios de manejo. Por ejemplo, ¿cuál es la probabilidad de extinción de una población en diez años, si se extraen 10 toneladas anuales de juveniles? Para responder este tipo de preguntas existen una serie de métodos de *Análisis de Viabilidad de Poblaciones* (PVA, por sus siglas en inglés). Veremos a en los siguientes párrafos un PVA basado en proyecciones de población, que son determinadas de matrices de transición con estocasticidad.

La estocasticidad es la variación causada por factores aleatorios. En el caso de las matrices de transición, existen diversos factores aportan incertidumbre a la estimación de la cantidad de individuos supervivientes en cada clase. Esto quiere decir, que la supervivencia en

cada celda de la matriz no es un valor fijo, sino, una variable aleatoria que tiene una distribución de probabilidad asociada.

Por ejemplo, suponga que en un muestreo se tiene que la etapa S_1 de una población se contaron un total de mil individuos, de los cuales cien pasan a la siguiente generación, seiscientos permanecen en dicha clase, y trescientos mueren. Entonces, las probabilidades de supervivencia son variables aleatorias asociadas a una distribución de probabilidad Beta⁵, y se puede representar como:

$$S_{1,1} \sim \text{Beta}(600, 400)$$

$$S_{1,2} \sim \text{Beta}(100, 900)$$

Este planteamiento permite obtener una infinidad de muestras de $S_{1,1}$ o $S_{1,2}$. Y asumiendo que el resto de etapas de la población hayan sido determinadas de la misma forma, entonces podemos obtener una infinidad de matrices de transición del conjunto de distribuciones de probabilidad. Cuando sea posible obtener muestras de la matriz de transición para una población, el paquete **popbio** ofrece métodos para estimar parámetros de población de manera estocástica.

Una vez cargado el paquete, la información sobre estas funciones se puede acceder usando: `?stoch.projection`, para realizar una proyección poblacional estocástica y obtener una aproximación de una distribución de probabilidad alrededor del tamaño poblacional; `?stoch.quasi.ext`, para analizar la probabilidad de extinción de una población a cierto número de años (o pasos de tiempo), bajo cierto umbral de individuos. Es decir, el investigador objetivamente indica un tamaño poblacional del cual se asume que la población no puede recuperarse, y el algoritmo calcula la probabilidad de que el tamaño de población se encuentre bajo este umbral.

El archivo `Source.R` provee una clase de referencia para los métodos estocásticos del paquete **popbio**. Esta clase lleva el nombre de `transStochMat`. Los argumentos para crear dicha clase son:

- **matriz**: una *lista* de matrices de transición. Se pueden generar a partir de las distribuciones de probabilidad asociadas a la estimación de los elementos de una matriz de transición.
- **n0**: El vector con el tamaño de población inicial.

⁵De hecho es más apropiada una distribución Multinomial; pero no entraremos en detalles.

- `t`: El valor máximo de los pasos de tiempo hacia los cuales se desea proyectar la población.
- `umbral`: El número mínimo de individuos bajo el cual la población se considera cuasi-extinta.
- `incluirEtapa`: Un vector de elementos binarios (unos y ceros), que indica cuales etapas o clases de edad deben incluirse en el análisis (unos), y cuales deben excluirse (ceros).

Para demostrar el uso de la clase de referencia, se utilizan los datos del paquete **popbio**. El conjunto de datos se llama *hudsonia* y ya provee una lista de matrices de transición.

```
# hudsonia es una lista de matrices de transición
data("hudsonia")
n <- c(4264, 3, 30, 16, 25, 5)
names(n) <-
  c("seed", "seedlings", "tiny", "small", "medium", "large")

hudsoniaStoch <- transStochMat$new(
  matriz = hudsonia,
  n0 = n,
  t = 50,
  umbral = 10, # Número de plantas con potencial reproductivo.
  incluirEtapa = c(0,1,1,1,1,1)
)
```

Una vez creado el objeto. Existen varias funciones asociadas para observar los resultados. Por ejemplo, para observar la distribución de probabilidad alrededor de la población (solo etapas seleccionadas en `incluirEtapa`) se utiliza el comando `hudsoniaStoch$plotN()` (Figura 3.4.4). La línea gris representa el tamaño de población inicial y la roja, el tamaño de población de cuasi-extinción.

Para observar la probabilidad de cuasi-extinción de la población (solo etapas seleccionadas en `incluirEtapa`), se digita `hudsoniaStoch$plotExtProb()` (Figura 3.4.5).

El parámetro de crecimiento con su incertidumbre asociada se obtiene así:

```
# GR estocástico
hudsoniaStoch$darGR()
```

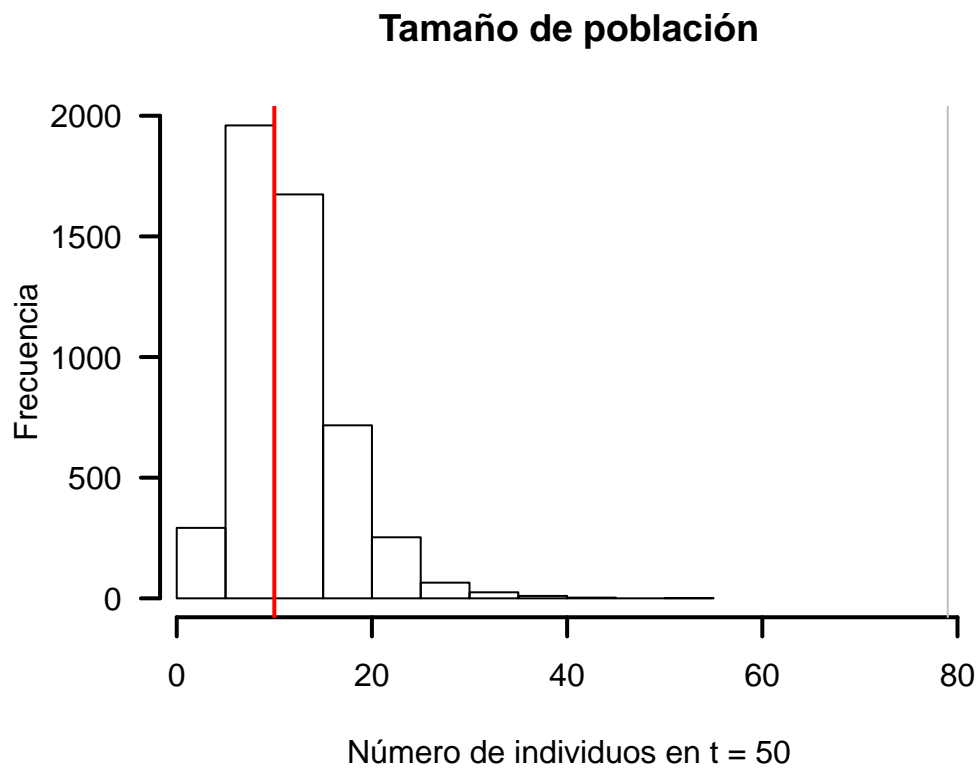


Figura 3.4.4: Población total de *hudsonia* a 50 años

```
## [1] Calculating stochastic growth at time 1
## [1] Calculating stochastic growth at time 10000
## [1] Calculating stochastic growth at time 20000
## [1] Calculating stochastic growth at time 30000
## [1] Calculating stochastic growth at time 40000
## [1] Calculating stochastic growth at time 50000

## $approx
## [1] 0.964
##
## $sim
## [1] 0.964
##
## $simCI
## [1] 0.963 0.964
```

Por último, la probabilidad de que la población se encuentre por debajo del umbral esta-

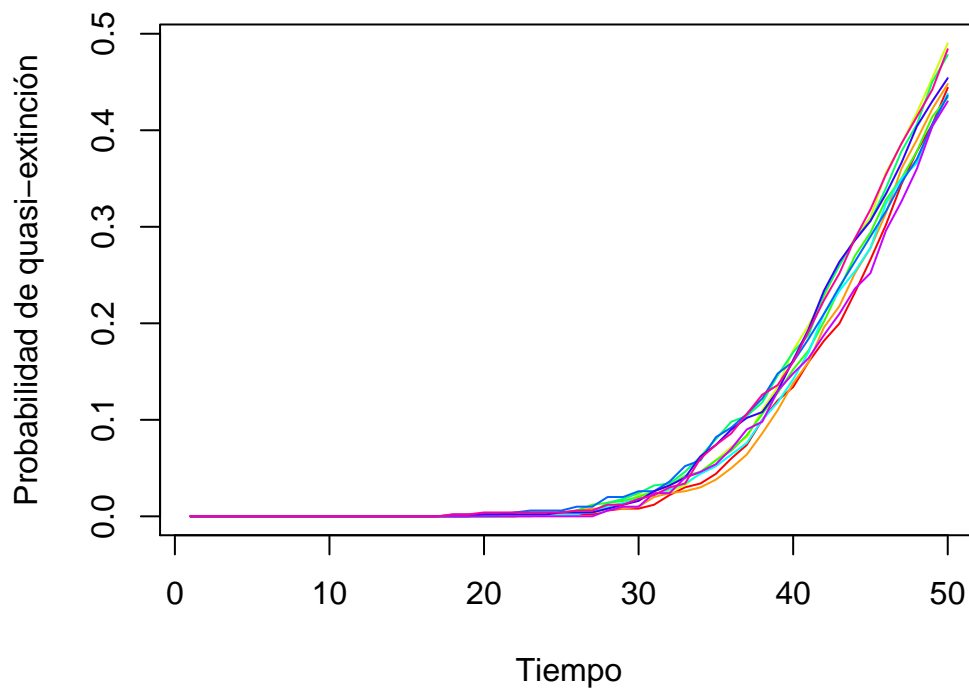


Figura 3.4.5: Probabilidad de extinción de *hudsonia* a 50 años. Umbral de quasi-extinción 10 individuos sobre tierra.

blecido, al finaliza el número de pasos de tiempo es:

```
hudsoniaStoch$extProb()
```

```
## [1] 0.45
```

3.5. Otras fuentes bibliográficas

En [Neal \(2004\)](#) se da una introducción gentil a la demografía, pero los métodos de análisis son limitados a unos pocos; básicamente, se limitan a enseñar las técnicas necesarias para comprender los fundamentos detrás de cada análisis. Por otro lado, [Skalski *et al.* \(2005\)](#) describe detalladamente las variantes de cada técnica, con un enfoque práctico; además, presenta discusiones sobre la utilidad de cada método de análisis y sus supuestos. Como nota precautoria, en [Skalski *et al.* \(2005\)](#), las convenciones sobre los nombres de las columnas de las tablas de vida difieren a las expuestas en este material.

Una pequeña reseña histórica, acompañada de una introducción gentil a los modelos de matrices se da en [Neal \(2004\)](#). Mientras que los detalles matemáticos de los cálculos relacionados con los modelos de matrices se da en [Skalski *et al.* \(2005\)](#). También se recomienda el uso de libros de Álgebra Lineal como referencia, al estudiar modelos matriciales.

Capítulo 4

Estimación de abundancia

El concepto de abundancia ha sido recurrente en los capítulos anteriores. También se le ha llamado *tamaño poblacional*, y usualmente se ha representado con la letra N . El tamaño de población, o abundancia, hace referencia a la cantidad de individuos que están delimitados de alguna forma. Esta delimitación puede ser geográfica, genética, o artificial. Una delimitación geográfica puede referirse a una extensión natural o ecosistema, o a un límite humano, como un parque nacional. Las delimitaciones genéticas, tienen que ver con el concepto de poblaciones con cierto grado de aislamiento reproductivo. Y la delimitación artificial puede referirse a una abundancia, por ejemplo, de *Daphnia* en un beaker, para pruebas de toxicología.

El problema, especialmente con la delimitación geográfica y genética, es que existe migración de individuos dentro y fuera de la población. Modelar la migración puede resultar muy complejo. A estos modelos de población se les conoce como *modelos de población abierta*. Por otro lado, si escogemos una escala temporal apropiada, podemos pensar en una población como *cerrada*, si asumimos que no existe ni emigración, ni inmigración dentro de la población.

En las siguientes páginas, cuando hablemos de abundancia, nos referiremos a poblaciones *cerradas*.

4.1. Abundancia no es un conteo

En algunos casos, los investigadores cometen el error de llamar abundancia, a la cantidad de individuos observados durante un muestreo. Este error tiene al menos dos componentes: Primero, los muestreos en general no abarcan toda el área de distribución de una población. Segundo, es poco probable que durante un muestreo se hayan observado todos los

individuos *sujetos a ser observados* por esa técnica de muestreo, es decir, la **probabilidad de detección** (o captura) es diferente de uno.

La probabilidad de detección es un parámetro central en las técnicas modernas de estimación de la abundancia. Usualmente, este parámetro se basa en modelos probabilistas, que tienen una distribución de probabilidad diferente de la Normal. Por tanto, las técnicas de análisis están basadas en los métodos de *máxima verosimilitud*, o *estadística bayesiana* (Royle y Dorazio, 2008).

El principio es muy sencillo. Cada individuo durante un muestreo tiene dos estados: *detectado*, o *no detectado*. Por tanto, la cantidad de individuos que fueron observados, son una porción del total de individuos. Si tamaño poblacional es igual a N , entonces, podemos decir que:

$$N = \frac{y}{p}$$

Donde y es el número de individuos *observados*, y p es la probabilidad de detección. Otra manera más formal de expresar lo anterior, es que el conteo, y , es una variable aleatoria que tiene una distribución binomial, con parámetros N y p .

$$y \sim \text{Binomial}(N, p)$$

El problema de este modelo, es que debemos estimar dos parámetros (N , y p) de un solo conjunto de datos (y). Esto implica que la identificación de los parámetros puede ser confusa; ya que, el mismo valor del conteo puede provenir de diferentes combinaciones de N , y p . Por esta razón, existen varios protocolos de muestreo que buscan aportar más información al modelo estadístico, para que logre identificar correctamente ambos parámetros.

—**Muestreo a distancia:** Consiste en hacer un conteo de animales y reportar junto a cada animal observado, la distancia perpendicular al transepto de muestreo. Usualmente utilizado para el conteo de animales grandes, este muestreo corrige la probabilidad de detección, modelando el decaimiento de este parámetro con la distancia (es más difícil detectar un animal a distancias más largas). En R existe el paquete `rdistance`.

—**Historial de capturas/recapturas:** Se utiliza en animales marcados. Cada animal genera un historial de captura/recaptura, de acuerdo al número de ocasiones de muestreo. Estos modelos se basan en una extensión de la distribución de probabilidad Binomial, llamada distribución multinomial. Uno de los paquetes más usados es el `Rmark`. Existen otras maneras

de analizar historiales de captura para estimar la abundancia, [Royle y Dorazio \(2008\)](#), enseña como hacerlo utilizando métodos bayesianos, con el software R y winBugs.

—**Muestreo por remoción:** El muestreo por remoción estima la probabilidad de captura (porque implica la captura de los individuos), basado en la proporción de individuos que son capturados de una ocasión de remoción a la siguiente. Por ejemplo, si en la primer remoción capturo 100 individuos, en la segunda capturo 50 individuos, y en la tercera capturo 25 individuos, la probabilidad de captura es $p = 0.5$; porque en cada ocasión de muestreo, capturo la mitad de los individuos de la ocasión anterior. Para llevar a la práctica los muestreos por remoción, los animales capturados en cada ocasión se separan del resto, ya sea al mantenerlos cautivos mientras dura el muestreo, o haciendo una marca para identificar que ya han sido capturados. En el último caso, los individuos re-capturados, no son tomados en cuenta en las próximas remociones. El paquete [unmarked](#) contiene métodos para muestreo por remoción.

—**Doble observador:** Este es una versión que mezcla los modelos de remoción, con los de captura y recaptura. El muestreo consiste en dos observadores, el principal anota todas sus observaciones, y las dice al observador secundario. Luego, el secundario anota, cualquier otro individuo que no fue visualizado por el observador primario. Con estos datos, se calcula la probabilidad de detección, y se estima la abundancia. El paquete [unmarked](#) contiene métodos para muestreo por doble observador.

Ejemplo 7 (Estimación del tamaño poblacional) *Se va a demostrar el principio de máxima verosimilitud para estimar el tamaño de población de un punto de muestreo. Los datos conocidos son el número de animales observados, y la probabilidad de captura.*

Primero, vamos a simular los datos. La simulación de datos es importante para validar métodos de análisis; ya que, permite conocer el valor de los parámetros reales, que luego serán estimados bajo el método a evaluar. Es decir, el primer paso en una simulación es establecer el modelo generador de datos, el cual debe contener los parámetros que deseamos estimar (o deben derivarse a partir de este modelo). El segundo paso, es utilizar los datos generados como entrada para el análisis, y estimar los parámetros deseados.

Generación de datos— El modelo es una simple generación de números aleatorios. Vamos a asumir, que se hicieron 5 repeticiones en el punto de muestreo. Fijamos el valor de los parámetros generadores en $N = 100$, y $p = 0.3$. Es decir en el sitio de muestreo existen 100 animales, pero solo logramos detectar aproximadamente el 30 % cada vez.

```
set.seed(1937)
y <- rbinom(n = 5, size = 100, prob = 0.3)
```

```
# n es el número de repeticiones, size = sería el tamaño poblacional
# y prob la probabilidad de captura
```

```
y
```

```
## [1] 32 30 34 28 35
```

Estimación de datos— La estimación de datos por máxima verosimilitud, implica encontrar una función que al ser optimizada (maximizada o minimizada), devuelva el valor del parámetro que maximiza la probabilidad de haber observado ese conjunto de datos. En este caso, nuestro modelo generador, es el mismo que usaremos en la estimación de datos. Esto no siempre es el caso en las simulaciones.

Asumimos que los conteos provienen de una distribución Binomial. También vamos a suponer que conocemos la probabilidad de detección ($p = 0.3$). Por tanto, lo que resta por hacer es buscar una función que devuelva la probabilidad¹ de haber observado esos datos, en función de N .

Suponiendo que los datos son independientes, la probabilidad de haber observado todo el conjunto es igual al producto del valor de probabilidad de observar los datos bajo una distribución binomial con p conocido, y N desconocido:

$$L(N) = \prod_{i=1}^n \text{Binomial}(y_i|N, p)$$

Al trabajar con el producto de probabilidades, es mejor convertir los valores a logaritmos, para evitar transgredir la barrera de precisión del computador. Entonces, cambiamos la función anterior por una *log-verosimilitud* ($lL(N)$). Además, dado que R por defecto minimiza las funciones, en vez de maximizarlas, vamos a *negar* la log-verosimilitud, para convertir nuestra optimización en un problema de minimización: $nL(N) = (-1) \times lL(N)$.

$$nL(N) = (-1) \times \sum_{i=1}^n \log(\text{Binomial}(N|y_i, p))$$

Ahora, nos toca traducir la expresión anterior, a una función en R. La ventaja es que tenemos una función ya construida para la probabilidad binomial, que toma como argumento si queremos los resultados en logaritmo: `dbinom(..., log=TRUE)`. Así que:

¹o un valor proporcional a la probabilidad

```
# Función de 'menos log-verosimilitud'
nlL <- function(p){#p, se refiere a parámetros.
  N <- floor(p[1]) #floor toma el entero
  valor <- -1*sum( dbinom(y, size = N, prob = 0.3, log = TRUE) )

  # A veces trabajar con logaritmos genera valores irreales.
  # Solo queremos valores válidos. El paso de abajo hace esto.
  valor <- valor[which(!is.na(valor) | !is.nan(valor) | valor != -Inf)]
  return(valor)
}
```

Una vez que hemos escrito la función, es hora de incluirla en el proceso de minimización, para encontrar el valor de N que hace más probable el haber observado esos datos:

```
salida <-
  optim(
    par = c(mean(y) / 0.3),
    fn = nlL,
    method = "Brent",
    lower = 0,
    upper = 1e6
  )
```

```
salida
```

```
## $par
## [1] 106
##
## $value
## [1] 13.1
##
## $counts
## function gradient
##      NA      NA
##
## $convergence
## [1] 0
```

```
##  
## $message  
## NULL
```

Primero, hay que entender la función `optim`. Su objetivo es tomar una función que devuelve *un solo valor*, y busca el mínimo de este valor. Para ello, `optim` explora el espacio de los parámetros que componen esa función y por diferentes métodos, devuelve la combinación de parámetros que minimiza la función. En este caso, solo tenemos un parámetro, por lo que es un problema de una sola dimensión. El argumento `par`, es un vector con los valores iniciales de los parámetros que queremos encontrar. En este caso, un buen valor inicial es el promedio del conteo, dividido por la probabilidad de detección que ya conocemos. `fn` es el objeto que contiene la función a minimizar. `method` es un argumento que requiere el nombre del método de optimización a utilizar. Para el método "Brent", se requiere de un valor mínimo (`lower`) y un máximo (`upper`), que delimitan el espacio del parámetro.

Vemos que el valor estimado del tamaño poblacional, $\bar{N} = 106.302$, se aproxima mucho al valor real que generó los datos $N = 100$. Existen formas de calcular la incertidumbre alrededor de estas estimaciones, pero están fuera del alcance del material. Sin embargo, vale la pena mencionar que la función de verosimilitud es proporcional a la distribución de probabilidad el parámetro que nos interesa (Figura 4.1.1. Esto nos permite utilizar métodos de remuestreo para obtener un conjunto de valores de \hat{N} , a los cuales se les pueden calcular los percentiles apropiados (intervalos de confianza).

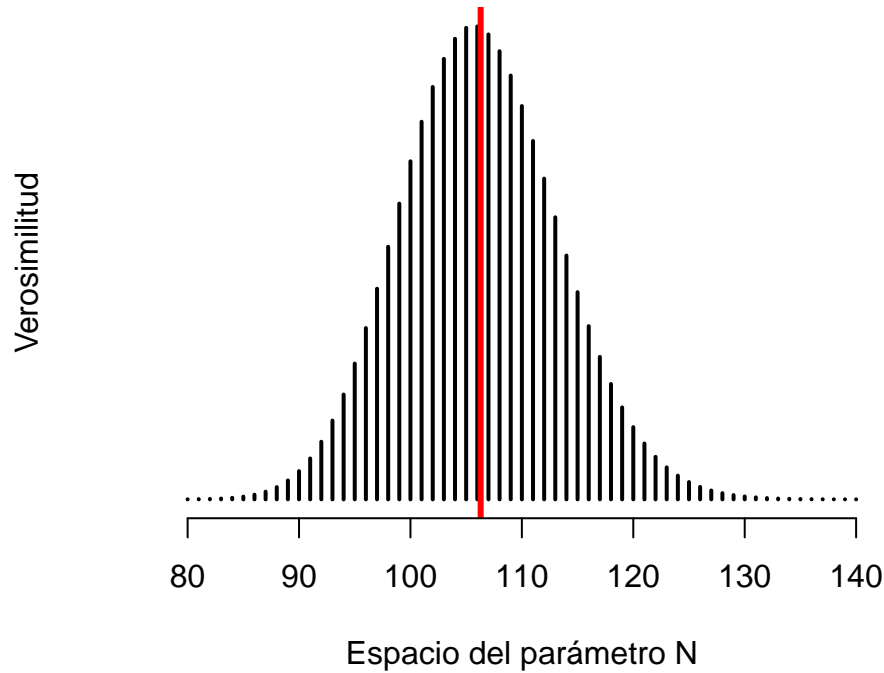


Figura 4.1.1: La verosimilitud es proporcional a una distribución de probabilidad

Ejemplo 8 (Principios del método de remoción) *Utilizando el método de máxima verosimilitud, se extenderá el modelo anterior para incluir un protocolo de muestreo por remoción, y estimar el tamaño poblacional y la probabilidad de captura.*

Generación de datos— La remoción consiste en muestrear un área de manera repetida, y remover los animales muestreados. De manera, que en cada repetición, solo se cuentan los animales nuevos. Esto quiere decir, que en cada remoción el tamaño de población cambia. Se inicia con una población N en el primer muestreo, pero para el segundo la población es igual a $N - y_1$. Para la tercer remoción, la población es igual a $N - y_1 - y_2$. Y así sucesivamente.

El modelo para generar datos entonces es:

$$y_i \sim \begin{cases} \text{Bin}(N, p), & k = 1 \\ \text{Bin}(N - \sum_{i=1}^k y_i, p), & k > 1 \end{cases}$$

Donde k es el k -ésimo evento de remoción.

```

set.seed(1934)
N <- 100
p <- 0.30
n <- 5# número de remociones
y <- numeric()

for ( k in 1:n){
  if (k==1){
    y. <- rbinom(n = 1, size = 100, prob = p)
    y <- append(y,y.)
  } else {
    y. <- rbinom(n = 1,size = 100 - sum(y), prob = p)
    print(sum(y))
    y <- append(y,y.)
  }
}

```

```

## [1] 28
## [1] 48
## [1] 60
## [1] 69

```

```

# n es el número de repeticiones, size = sería el tamaño poblacional
# y prob la probabilidad de captura

```

```

cat("el vector de conteos es ",y,"\n")

```

```

## el vector de conteos es 28 20 12 9 8

```

Estimación de datos— Nuestro modelo de verosimilitud debe adaptar ahora, la nueva información generada por las remociones. El punto clave, es que en cada remoción el tamaño de la población disminuye de acuerdo a la cantidad de individuos que fueron removidos. Para ello, podemos calcular un vector auxiliar, que contiene el número acumulado de individuos removidos en cada muestreo:

$$c_i = \begin{cases} 0, & k = 1 \\ \sum_{i=1}^k y_i, & k > 1 \end{cases}$$

En R:

```
c <- 0

c <- append(c,cumsum(y), after = 1)

c <- c[-length(c)] # último elemento no se usa

c
```

```
## [1] 0 28 48 60 69
```

Ahora nuestra función de verosimilitud es:

$$nLL(N, p) = \sum_{i=1}^n \log(\text{Binomial}(y_i | N - c_i, p))$$

Nótese la inclusión de c en la función. Ahora, podemos tomar la función que escribimos en el ejemplo 7, y modificarla de acuerdo a este nuevo parámetro:

```
# Función de 'menos log-verosimilitud'
nLL <- function(p,c){#p, se refiere a parámetros.
  N <- floor(p[1])
  N <- ifelse(N <= max(c), max(c), N) # Si es menor al número acumulado de
                                     # individuos, lo fija al límite inferior.

  x <- p[2]
  probCap <- exp(x)/(1+exp(x)) # calcula un valor entre 0 y 1

  valor <- -1*sum( dbinom(y, size = N-c, prob = probCap, log = TRUE) )

  # A veces trabajar con logaritmos genera valores irreales.
  # Solo queremos valores válidos. El paso de abajo hace esto.
  valor <- valor[which(!is.na(valor) | !is.nan(valor) | valor != -Inf)]
  return(valor)
}
```

```
##Funcion generadora
gen <- function(p,c){#p, se refiere a parámetros.
  N <- floor(p[1])
  N <- ifelse(N <= max(c), max(c)+1, N)
  N <- N + sample(c(-1,1),size=1,prob = c(0.5,0.5))

  x <- p[2]
  x <- x + rnorm(1,0,0.2)
  return(p=c(N,x))
}
```

En este caso, vamos a utilizar una función generadora. Esta función genera los valores de manera aleatoria, pero con ciertas restricciones. Estas restricciones sirven para mantener el algoritmo de optimización dentro de un espacio del parámetro, que consideramos apropiado. De lo contrario, el algoritmo puede volverse *loco* y arrojar valores absurdos. La selección de la función generadora, requiere de conocimiento del modelo, y es un arte más que una ciencia.

Posteriormente, introducimos nuestra función de menos log-verosimilitud y la función generadora, en la rutina de optimización. Ahora escogemos un método más apropiado para estimar dos o más parámetros. Éste es el método “SANN”, que toma valores al azar del espacio de los parámetros, de acuerdo a las reglas de la función generadora. Para cada combinación de parámetros, calcula el valor de $nL(N, p)$, y guarda la combinación que tenga el menor valor. Recuerde que es un problema de minimización.

```
salida <-
  optim(
    par = c(max(c)*1.1,0),# valores iniciales
    fn = nll,
    gr = gen,
    c=c,
    method = "SANN",
    control = list(maxit=50000)
  )

salida
```

```
## $par
```



```
## [1] 91.000 -0.809
##
## $value
## [1] 10.5
##
## $counts
## function gradient
##      50000      NA
##
## $convergence
## [1] 0
##
## $message
## NULL
```

En este caso particular, tenemos que convertir a una escala apropiada para probabilidades, el segundo valor de `salida$par`. Para ello, utilizamos una función llamada *expit* que mapea los valores $\text{expit} : \{-\infty, \infty\} \mapsto \{0, 1\}$. La función *expit* es igual a $\exp(x)/(1 + \exp(x))$.

```
p_est <- exp(salida$par[2])/(1+exp(salida$par[2]))
p_est
```

```
## [1] 0.308
```

Vemos que el valor real de N es 100, y el valor estimado es 91. Mientras que el valor real de p es 0.30, y el valor estimado fue de 0.308.

Capítulo 5

Interacciones entre poblaciones

Desde el punto de vista de la dinámica de poblaciones, nos interesa una definición funcional sobre las interacciones entre especies. Modificando la definición en [Neal \(2004\)](#), las interacciones entre especies ocurren *cuando dos o más individuos o especies experimentan cambios en su fitness*. Esta definición deja abierta la posibilidad de que las interacciones sean positivas (como la simbiosis), o negativas (la depredación).

La mejor forma de entender esto es con un ejemplo. Considere una población de ratones silvestres, y una población de coyotes que depredan a estos ratones. En ausencia de los coyotes, los ratones crecerían de manera logística (ecuación (2.6)); sin embargo, en presencia de este depredador hay que agregar un nuevo término a esta relación para visualizar el efecto de los coyotes.

Este término es una constante que indica la cantidad de ratones que come *un solo* coyote por unidad de tiempo. Llamaremos a esta constante como β . Por tanto, al multiplicar β por el tamaño poblacional del coyote, obtenemos la cantidad de ratones que son depredados en total por unidad de tiempo. Es decir, el tamaño de población de ratones al finalizar el año es igual a $N_r - \beta N_c$. Donde N_r es el tamaño poblacional de los ratones y N_c el tamaño poblacional de los coyotes. Aplicando este nuevo principio, podemos redefinir la ecuación ((2.6)) como:

$$\frac{dN_r}{dt} = r_r N_r \left(\frac{K_r - N_r - \beta N_c}{K_r} \right)$$

Donde r_r es la tasa de crecimiento intrínseco de los ratones, y K_r es la capacidad de carga de la población de ratones.

El modelo no está completo aún, ya que las dos poblaciones ejercen efectos entre sí, y

hasta ahora solo hemos descrito el efecto de la población de coyotes, sobre la población de ratones. El efecto de las presas sobre los depredadores es positivo, es decir, entre mayor cantidad de presas, mayor cantidad de depredadores. Quiere decir, que la constante para el caso de los depredadores tendrá el signo opuesto de β .

Procediendo de manera similar, la población de coyotes es igual a $N_c + \alpha N_r$, donde el término αN_r es la cantidad de coyotes *adicionales* a la población, que se sostienen gracias a la población de ratones. Podemos modificar la ecuación de la tasa de crecimiento en coyotes de manera similar a la de ratones, cambiando los términos correspondientes. Y cuando juntamos las dos funciones, terminamos con un *sistema de ecuaciones diferenciales*, donde una ecuación depende de la otra.

$$\begin{aligned}\frac{dN_r}{dt} &= r_r N_r \left(\frac{K_r - N_r + \beta N_c}{K_r} \right) \\ \frac{dN_c}{dt} &= r_c N_c \left(\frac{K_c - N_c + \alpha N_r}{K_c} \right)\end{aligned}\tag{5.1}$$

Ejemplo 9 *Grafique el comportamiento de una población de ratones, y de coyotes. Proyecte la población a treinta años. Apóyese en el material del curso sobre métodos numéricos y la integración de Euler.*

Asuma que la población de ratones tiene los siguientes parámetros: $N_r(0) = 100$, $K_r = 1200$, $r_r = 0.49$, $\alpha = 1/65$. Para los coyotes, los parámetros son: $N_c(0) = 100$, $K_c = 30$, $r_c = 0.14$, $\beta = -60$.

En R, se puede modelar la ecuación (5.1) de la siguiente forma:

```
plotLV <- function(N1 = 100,
  N2 = 50,
  alpha = 1/65,
  beta = -60,
  r1 = .49,
  r2 = .14,
  K1 = 1200,
  K2 = 30,
  t = 30,
  dt = 1 / 12) {
```

```
pop1 <- numeric()
pop2 <- numeric()
timeLine <- numeric()

pop1[1] <- N1
pop2[1] <- N2

i <- 0
timeLine <- i
while (i < t) {
  N1. <- pop1[length(pop1)]
  N2. <- pop2[length(pop2)]

  N1. <- N1. + r1 * N1. * ((K1 + beta * N2. - N1.) / K1) * dt
  N2. <- N2. + r2 * N2. * ((K2 + alpha*N1. - N2.) / K2) * dt

  pop1 <- append(pop1, N1.)
  pop2 <- append(pop2, N2.)

  i <- i + dt
  timeLine <- append(timeLine,i)
}

par(oma = c(2,2,1,2))
plot(
  c(0, t),
  range(c(pop1, pop2)),
  axes = F,
  type = "n",
  xlab = "tiempo",
  ylab = "Tamaño de población")

lines(timeLine,pop1, lty = 1)
axis(side = 1)
axis(side = 2)
par(new = TRUE)
```

```
plot(
  timeLine,
  pop2,
  lty = 2,
  axes = FALSE,
  xlab = "",
  ylab = "",
  type = "l")

axis(side = 4)
mtext("Coyote", side = 4, line = 2)
legend("right",lty = c(1,2), legend = c("presa","predador"))
abline(h = 0, lwd=3)
box(which = "plot")
}
```

Y para graficar la interacción entre especies, solo llamamos la función.

```
plotLV()
```

```
## [1] 50.0 49.6 49.3 48.9 48.6 48.3
```

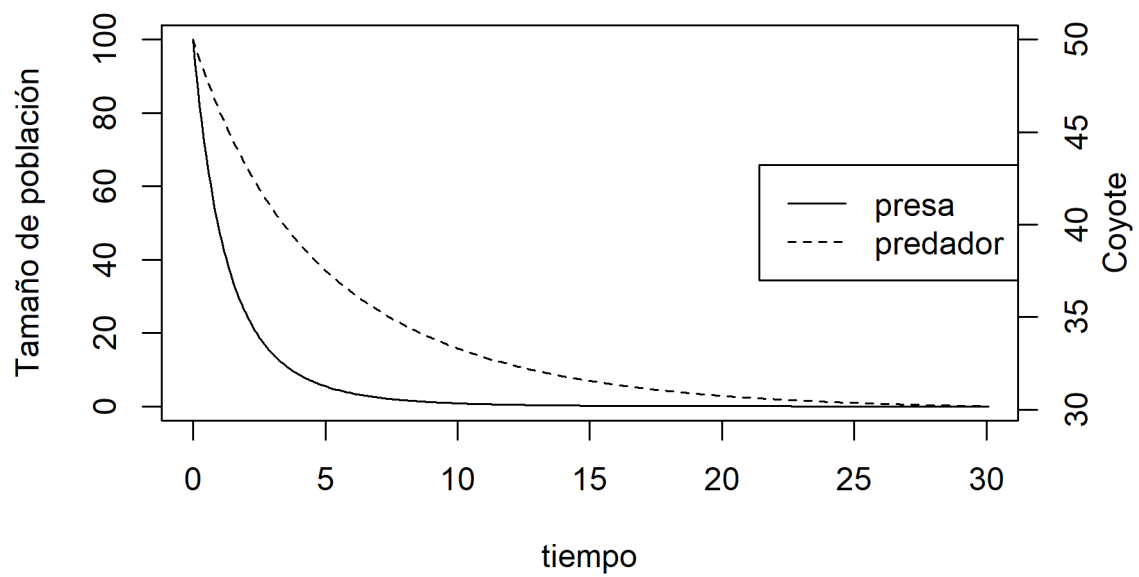


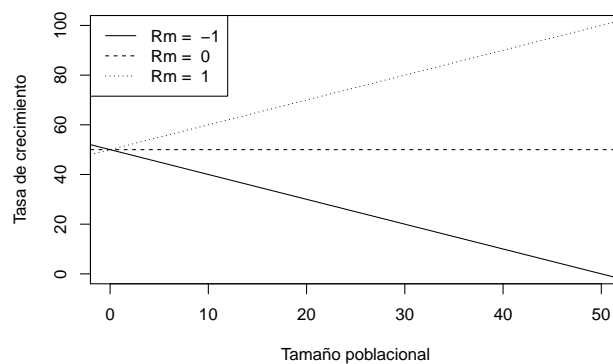
Figura 5.0.1: Interacción depredador-presa, entre una población hipotética de ratones, y una población hipotética de coyotes.

Capítulo 6

Soluciones a los ejercicios

Ejercicio 1

```
plot(0,0,xlim = c(0,50),  
     ylim = c(0,100) ,  
     type = 'n',xlab = 'Tamaño poblacional',  
     ylab = "Tasa de crecimiento")  
  
val=numeric()  
  
for(Rm in c(-1,0,1)){  
  abline(50,Rm,lty=Rm+2)  
  val=append(val,Rm)  
}  
legend("topleft",lty= 1:3,legend = paste('Rm = ',val))
```



R_m representa la pendiente de la recta.

Ejercicio 5

En 3 horas existen 12 periodos de 15 minutos ($t = 12$). Entonces aplicamos la ecuación (2.3):

$$100 \times 2^{12} = 409\,600 \text{ células}$$

Ejercicio 6

Obtener $r_m = \ln \lambda$; luego multiplicar $r_{mes} = r_m \times 4$ para obtener la escala a meses. Finalmente, transformar $\lambda_{mes} = e^{r_{mes}}$.

Ejercicio 8

Asumimos que r_m es cualquier constante positiva ($r_m \in \mathbb{R}^+$). Entonces $-1 \times r_m \in \mathbb{R}^-$.

Luego, tomamos el límite:

$$\lim_{t \rightarrow \infty} N_0 e^{-r_m t}$$

Que equivale a:

$$\lim_{t \rightarrow \infty} \frac{N_0}{e^{r_m t}}$$

Vemos que el denominador de la expresión anterior es un número que crecerá infinitamente. Si reemplazamos $e^{r_m t}$ por x , cuando $t \rightarrow \infty \Rightarrow x \rightarrow \infty$. Y quedamos con la expresión:

$$\lim_{x \rightarrow \infty} \frac{N_0}{x} = 0$$

Porque cuando $x \rightarrow \infty \Rightarrow x \gg N_0$. Es decir, cuando x se vuelve infinito, es mucho más grande que N_0 , por tanto, el cociente tiende a cero, cuando x tiende a infinito.

Poblacionalmente, esto significa, que si una población mantiene una tasa intrínseca negativa, por un periodo de tiempo suficientemente largo, sufrirá un evento de extinción.

Ejercicio 11

El tamaño de población que maximiza el crecimiento

Reconocemos que la capacidad de carga son $K = 100$ kg. Por tanto, la cantidad de biomasa que maximiza el crecimiento, según el *máximo rendimiento sostenible* es $K/2 = 50$ kg.

¿Cuánta biomasa se debe extraer en un día para mantener la tasa de crecimiento al máximo?

Debemos utilizar unidades congruentes durante los cálculos. Así que el primer paso convertir r_m de s^{-1} a d^{-1} .

$$\frac{5 \times 10^{-6}}{s} \frac{86400 s}{d} = \frac{0.432}{d}$$

Ahora podemos calcular el *Máximo Rendimiento Sostenible*, reemplazando los valores apropiados en:

$$MRS = \frac{r_m K}{4} = 10.8 \text{ kg}$$

¿A qué biomasa total se debe cosechar el tanque?

Esta es la suma del tamaño de población al *máximo rendimiento sostenible*, más el *máximo rendimiento sostenible*:

$$\frac{K}{2} + \frac{r_m K}{4} = 60.8 \text{ kg}$$

Ejercicio 16

```
M <- matrix(c(
  5,10,15,20,
  20,45,10,5,
  0,0,1.2, 1.1
), ncol = 3,
byrow = FALSE)

colnames(M) <- c("x", "Sx", "mx")

lf <- tablaVida(x.sup=M[, "x"], Sx=M[, "Sx"], mx=M[, "mx"])

re <- log(lf$R0)/lf$Tc
```

```

if (re > 1){
  print("Está creciendo")
} else if (re == 1){
  print("Se mantiene constante")
} else if ( re < 0 ){
  print("Está decreciendo")
}

```

```
## [1] "Está decreciendo"
```

```
rm(M)
```

Ejercicio 17

Cuadro 6.0.1: Tabla de vida para *Ovis dalli*.

x	S_x	D_x	l_x	d_x	q_x	e_x
1	608	121	1.000	0.199	0.199	7.071
2	487	7	0.801	0.012	0.014	7.703
3	480	8	0.789	0.013	0.017	6.808
4	472	7	0.776	0.012	0.015	5.915
5	465	18	0.765	0.030	0.039	4.997
6	447	28	0.735	0.046	0.063	4.178
7	419	29	0.689	0.048	0.069	3.424
8	390	42	0.641	0.069	0.108	2.641
9	348	80	0.572	0.132	0.230	1.899
10	268	114	0.441	0.188	0.425	1.317
11	154	95	0.253	0.156	0.617	0.922
12	59	55	0.097	0.090	0.932	0.602
13	4	2	0.007	0.003	0.500	1.000
14	2	2	0.003	0.003	1.000	0.500
15	0	NA	0.000	NA	NA	NaN

Ejercicio 18

```

op <- par()
par( mar = c(5, 5, 4, 5) + 0.1)

plot(lf[,1],lf[,4]*1000,
     xlab = "Edad", ylab = "", axes = FALSE, type = "n")

```

```

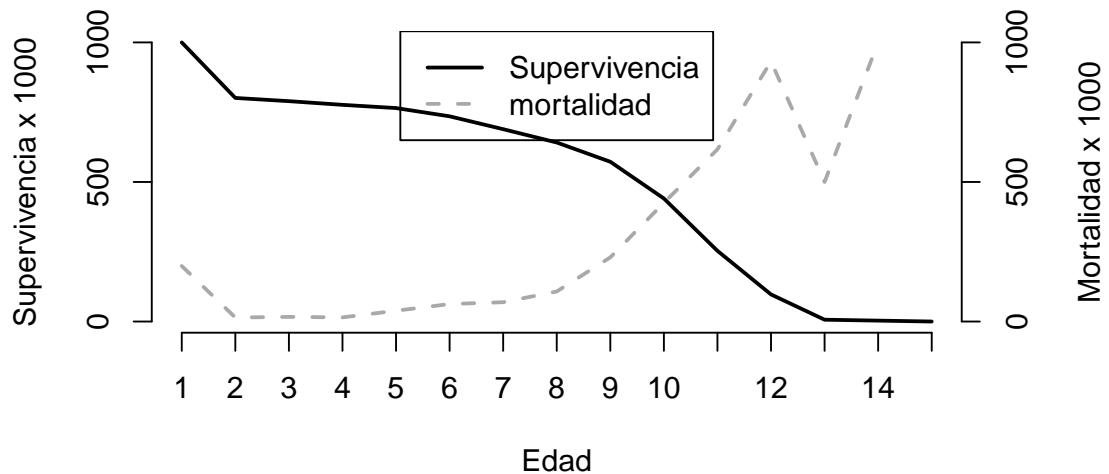
axis(side = 1, at = lf[,1])

axis(side = 2, at = c(0,500,1000))
lines(lf[,1],lf[,4]*1000, lwd = 2)
mtext(side = 2, "Supervivencia x 1000", line = 3)

axis(side = 4, at = c(0,500,1000))
lines(lf[,1],lf[,6]*1000, lwd = 2, lty = 2, col = "darkgray")
mtext(side = 4, "Mortalidad x 1000", line = 3)

legend("top", legend = c("Supervivencia", "mortalidad"),
      lty = c(1,2), lwd = c(2,2), col = c(1, "darkgray"))

```



```

par(op)# restaurar valores por defecto para plot
rm(lf)#limpiar un poco

```

Según el gráfico anterior, existe un primer pico de mortalidad en las crías de un año o menos. Posteriormente la mortalidad es muy baja hasta los ocho años, cuando empieza a incrementar. Probablemente, a partir de los ocho años, las cabras de monte empiezan a envejecer y a perder sus cualidades para sobrevivir.

Índice alfabético

A

abundancia, [47](#)

C

crecimiento, [7](#)

crecimiento denso-independiente, [8](#)

crecimiento geométrico, [8](#)

cohorte, [23](#)

D

dinámica de poblaciones, [5](#)

E

ecología de poblaciones, [5](#)

I

interacciones entre especies, [59](#)

M

matrices de transición, [34](#)

 multiplicación matricial, [35](#)

P

probabilidad de detección, [48](#)

T

tamaño de población, [47](#)

tasa de crecimiento, [8](#)

tasa de crecimiento neta, [24](#)

tasa de multiplicación, [9](#)

tasa instantánea de crecimiento, [12](#)

Apéndice A

Métodos numéricos para ecología de poblaciones

A.1. Simulación de ecuaciones diferenciales

Tomaremos el ejemplo de la ecuación (2.4), para mostrar un método para encontrar el tamaño de población, sin tener que utilizar cálculo (Barrantes Campos, 2015). Este es el *método de Euler*, que se explicará mediante un ejemplo.

Una derivada implica un cambio infinitesimal de una variable en relación a otra. Por ejemplo, el cambio en el número de individuos de una población en un momento pequeñísimo de tiempo, puede representarse como la diferencia de la población, entre la duración de ese pequeño intervalo de tiempo:

$$\frac{dN}{dt} \approx \frac{\Delta N}{\Delta t} = \frac{N_t - N_{t-\Delta t}}{\Delta t}$$

Si sustituimos en la ecuación (2.5), tenemos:

$$\frac{N_t - N_{t-\Delta t}}{\Delta t} = r_m N$$

Arreglando la expresión anterior, podemos despejar en terminos de N_t :

$$N_t = N_{t-\Delta t} + r_m N_{t-\Delta t} \Delta t$$

Hay que resaltar que esta **no** es una solución exacta; sino, una aproximación. Entre más pequeño se haga Δt , más se aproximará el resultado, al valor exacto dado por (2.5). En casos donde no existe una solución analítica, o simplemente, no es sencillo resolver la ecuación, siempre se puede recurrir a los métodos numéricos, para tener una idea de la solución real.

Para programar este sencillo ejemplo, necesitamos varios pasos:

- Definir un valor inicial de la población, y el valor de t en el cuál queremos conocer el tamaño de población.
- Definir el Valor de r_m .
- Establecer un criterio para guardar el valor de N_t , cada cierto lapso de tiempo. (Para no crear un objeto virtual innecesariamente grande)
- Crear un objeto para guardar el tamaño de la población, y los puntos de tiempo a los que está asociada.
- Definir el tamaño de Δt , y calcular el número de iteraciones necesarias hasta llegar al final del periodo de tiempo de interés.
- Crear un bucle, y ejecutar iterativamente la integración de Euler.
- Definir un criterio para detener el algoritmo.

El siguiente algoritmo generaliza todas las funciones dependientes de $N_{t-\Delta t}$.

$$N_t = N_{t-\Delta t} + f(N_{t-\Delta t}, \mathbf{c}) \Delta t$$

Donde \mathbf{c} son constantes.

```
euler <- function(
  fooName,
  valInic,
  tiempoParar,
  NoIter,
  guardarCada,
  ...) {
  arg <- list(...)
  fn <- get(fooName)

  #Encuentra los argumentos provistos
```



```

argName <- match.arg(names(arg), #arg provistos
formalArgs(fn), #arg existentes
several.ok = TRUE)
#Nombra la lista con los nombres de los argumentos provistos
names(arg) <- argName

deltaT <- tiempoParar/NoIter

val <- numeric()
val[1] <- valInic

valTmp <- numeric()
valTmp <- val[1]

#Completa la lista de argumentos con N[t-1]
arg[[length(arg) + 1]] <- valInic
totalArg <- length(arg)
#Escribe todos los nombres de los argumentos, para do.call
names(arg) <-
formalArgs(fn)#Encuentra los nombres de los argumentos

tiempo <- numeric()

counter <- 0
tNow <- 0
tiempo[1] <- 0

while (tNow < tiempoParar) {
  valTmp <- valTmp + do.call(fn, args = arg) * deltaT
  tNow <- tNow + deltaT
  arg[[totalArg]] <- valTmp
  counter <- counter + 1

  if (counter == guardarCada) {
    val <- append(x = val, values = valTmp)
    tiempo <- append(x = tiempo, values = tNow)
  }
}

```

```
    counter <- 0
  }
}

return(list(
  poblacion = val,
  tiempo = tiempo,
  tNow = tNow,
  arg = arg,
  Dt = deltaT
))
}
```

Por ejemplo:

```
diffG1 <- function(rm, N) N * rm
N0 <- 10

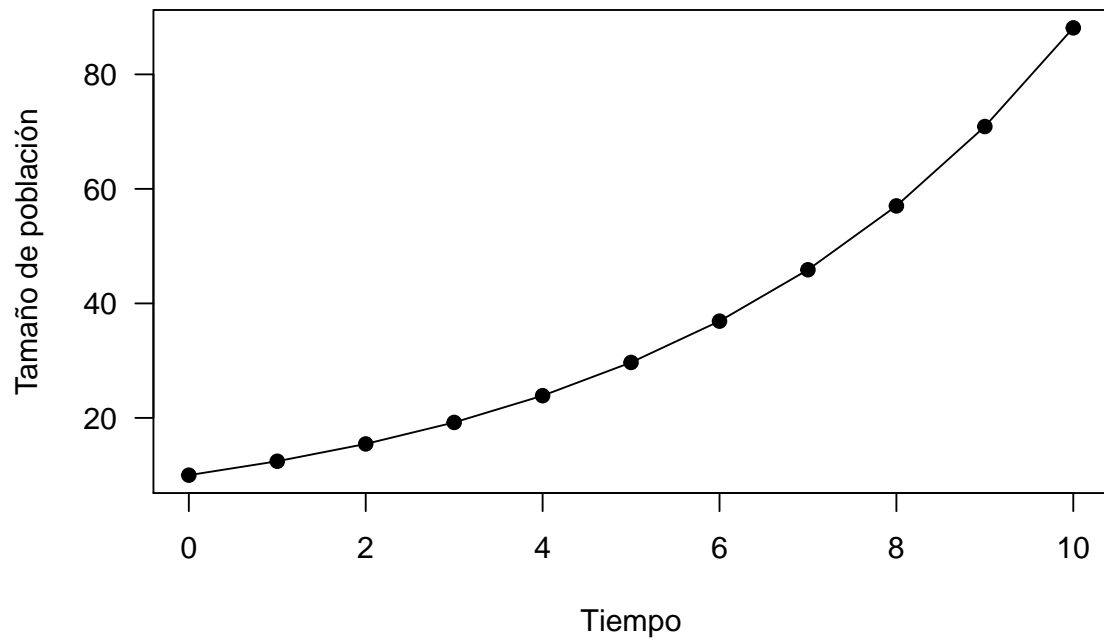
Resultados1 <- euler(fooName = "diffG1", valInic = N0, tiempoParar = 10, NoIter = 100,
  guardarCada = 10, rm = 0.22)

diffG2 <- function(rm, Kmax, N) rm * (1 - N/Kmax)

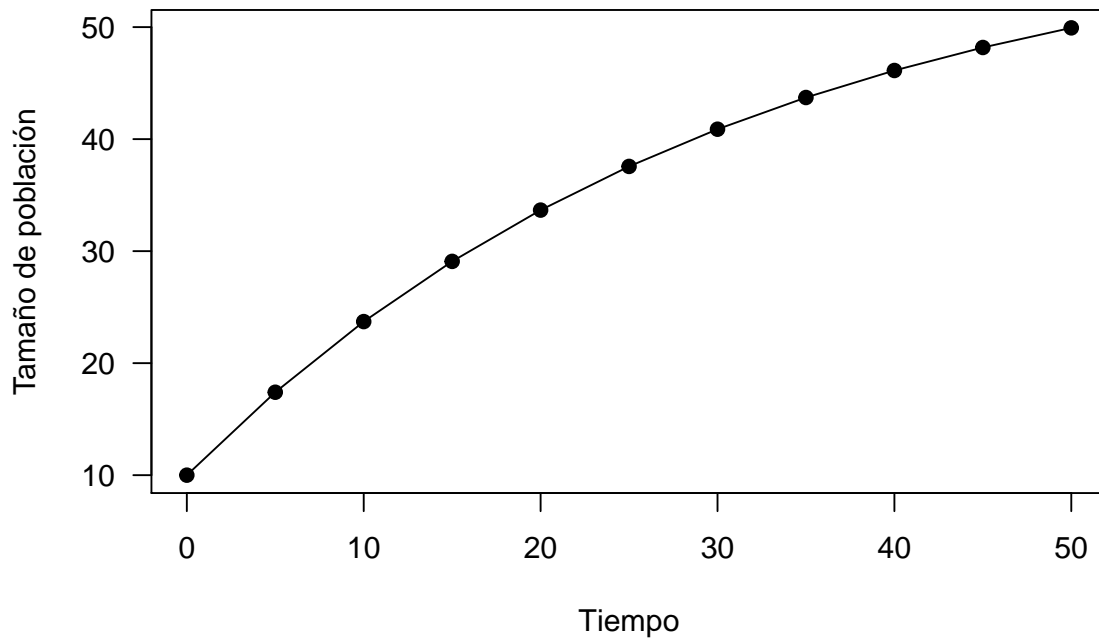
N0 <- 10

Resultados2 <- euler(fooName = "diffG2", valInic = N0, tiempoParar = 50, NoIter = 500,
  guardarCada = 50, rm = 1.92, Kmax = 60)

plot(Resultados1$tiempo, Resultados1$poblacion, type = "p", xlab = "Tiempo",
  ylab = "Tamaño de población", las = 1, pch = 21, bg = 1)
lines(Resultados1$tiempo, Resultados1$poblacion)
```



```
plot(Resultados2$tiempo, Resultados2$poblacion, type = "p", xlab = "Tiempo",  
      ylab = "Tamaño de población", las = 1, pch = 21, bg = 1)  
lines(Resultados2$tiempo, Resultados2$poblacion)
```

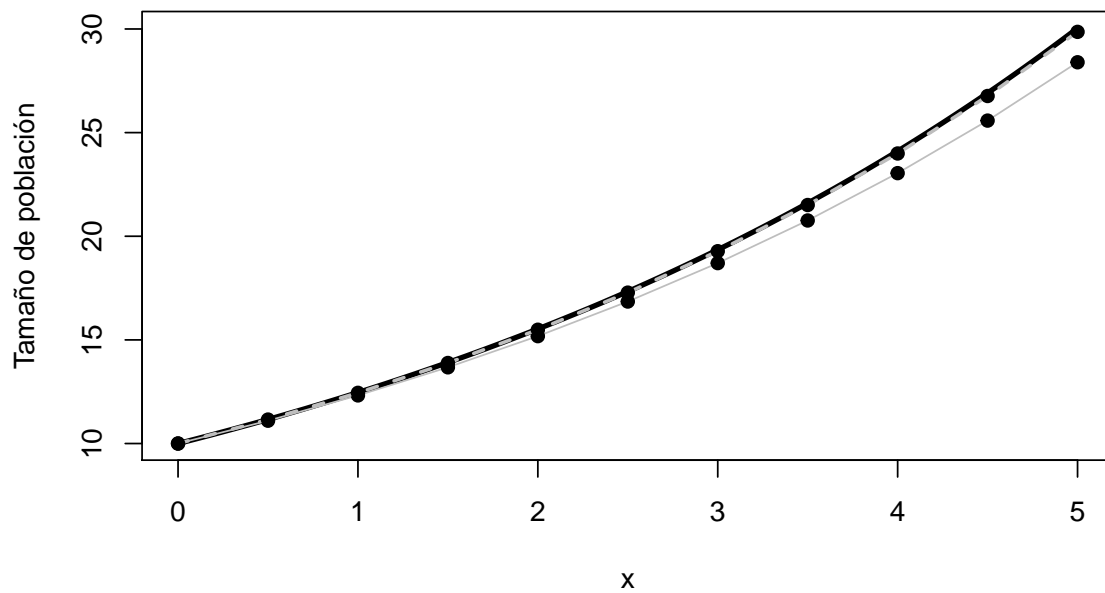
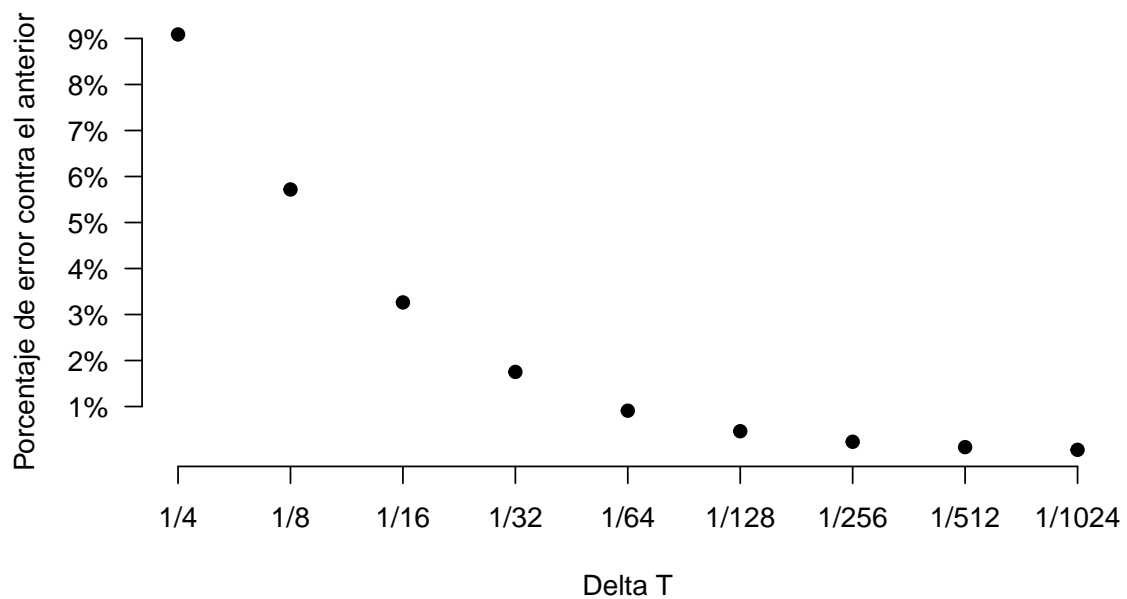


A.2. Error en las simulaciones

Hemos mencionado que la integración de Euler es una *aproximación* a la función verdadera. Que tan bien o mal sea esta aproximación depende de la elección de Δt .

Por ejemplo en la figura [A.2.1](#) la línea negra representa la ecuación verdadera, la línea gris continua representa una aproximación con $\Delta t = 1/2$, y la línea gris discontinua representa una aproximación con $\Delta t = 1/100$. Observamos que la distancia de la línea con $\Delta t = 1/100$ con la curva real es menor que la distancia de la línea con $\Delta t = 1/2$ a la curva real. Es decir, si disminuimos el valor de Δt disminuimos el error de la aproximación [A.2.2](#).

En este caso, solo se calculó el error para el valor de $N_{t=5}$.

Figura A.2.1: Comparación del efecto del tamaño del Δt Figura A.2.2: Porcentaje de error para N en $t = 5$ según el Δt

Apéndice B

Tutorial de R con RStudio

B.1. Crear un proyecto en *RStudio*

Crear un proyecto en **RStudio** para cualquier proyecto con **R**, es importante. Los proyectos organizan los documentos en una sola carpeta, y son fundamentales para el control de versión con un software como **Git**.

Abrimos **RStudio**, y ubicamos la barra de herramientas en la parte superior. El primer paso es ir a `file --> New Project`. Creamos una carpeta en una ubicación que nos permita tener derechos de administrador, e idealmente, fuera de cualquier carpeta de sincronización en línea¹.

RStudio nos guiará por los siguientes pasos:

- **Crear un Proyecto:** Escogemos que sea un nuevo directorio.
- **Tipo de Proyecto:** Escogemos *nuevo proyecto*.
- **Crear:** Escogemos la carpeta, y el nombre del proyecto. **NO** marcamos *crear repositorio con Git*.

Volvemos a la barra de herramientas, en **RStudio**, y vamos a `File --> New file --> R script`. Este archivo solo soporta código en **R**, con la gran ventaja de que colorea las funciones, variables y estructuras más comunes; lo cual, hace que el código sea más legible.

IMPORTANTE: En el siguiente tutorial, el código en **R** se encuentra dentro de ambientes

¹En caso de que quieran tener un respaldo en la nube, se recomienda pausar la sincronización mientras se trabaja en el proyecto

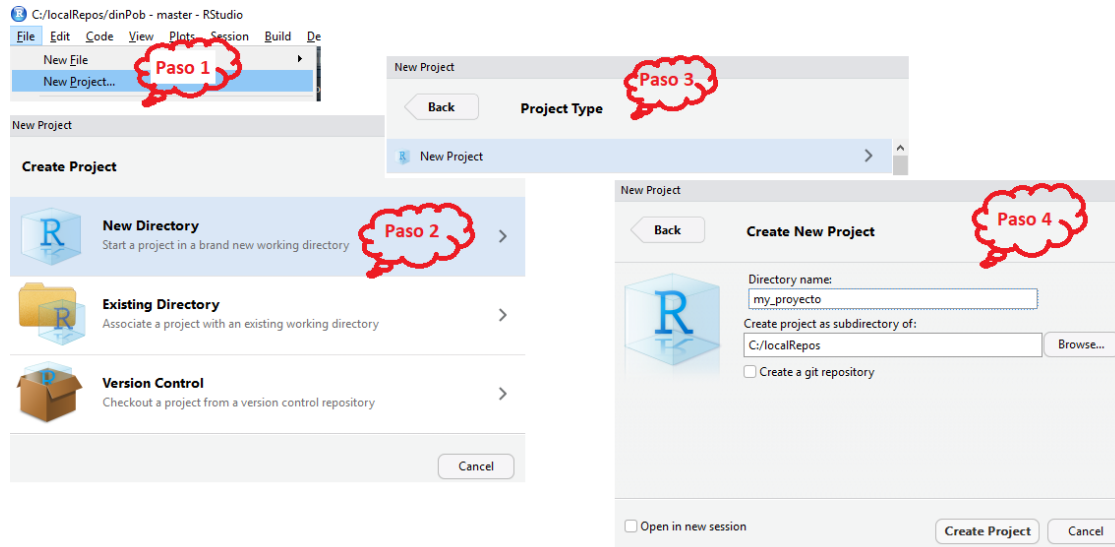


Figura B.1.1: Cómo hacer un nuevo proyecto, en **RStudio**.

especiales, rodeados por una caja gris. Escribe las líneas que se muestran en esa caja, y para ver el resultado en la consola de **R**, apreta **Ctrl + ENTER**.

B.2. Funciones básicas en R

Ahora que hemos creado un proyecto, y tenemos un lienzo en blanco, empezamos por ver las funciones más elementales. **R** contiene todas las operaciones básicas como: adición, substracción, multiplicación, división, potencias, y logaritmos.

Tras cada línea presiona Ctrl + ENTER

`1 + 1` *# adición*

`1-1` *# substracción*

`1*2` *# multiplicación*

`1/2` *# división*

`2^(8)` *#potencia*

`log(2)` *# logaritmo natural*


```
log10(2) #logaritmo base 10

log(x = 2, base = <n>) #logaritmo base <n>, donde <n> se
                        # reemplaza por cualquier número.
```

Notar que en el código, cualquier línea de texto precedida de # es un comentario, que no será evaluado por el computador. Agregar comentarios es muy útil, si uno va a re-utilizar parte del código en otro momento. Ayuda a mantener la claridad en lo que se está haciendo.

Continuando con las operaciones básicas, también podemos mezclar operaciones de la forma que convenga. Siempre considerando las reglas de prioridad por paréntesis. Por ejemplo, si queremos calcular el logaritmo del resultado de una función, para una base 16.

```
log( x= 1 / (1 + (5/2) ) , base = 16 )
```

Luego, como la mayoría de lenguajes de programación, podemos asignar valores a un objeto y utilizarlo después en otra operación:

```
a <- log10(4/3) # 'a' tiene el valor de la operación

b <- a^2 # Equivale a log10(4/3)^2

c <- b^2 - a

c
```

Esto quiere decir que podemos crear un objeto con el operador <-, que pueden ser datos, o resultados de otras operaciones, para incluirlo en una nueva función. Por lo que la salida de una función puede ser la entrada de la próxima.

B.3. Estructuras de datos

R puede manejar objetos muy complejos; sin embargo, estos objetos generalmente se componen de partes muy sencillas. Revisaremos éstas partes sencillas, y luego crearemos un objeto más complejo.

Vectores

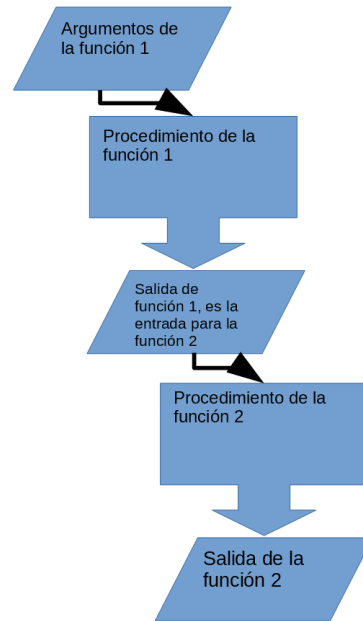


Figura B.2.1: Flujo de entradas y salidas en una función de R

Un vector en programación, es una colección de uno o más valores. Podemos pensar que un vector en R equivale a una matriz de n filas, y solo una columna.

```
vect1 <- c(1,2,3,4,5,6,7,8,9,0) #es una concatenación de
# números, que se crea con la función 'c(...)'

vect2 <- rnorm(10) # son diez números al azar obtenidas
# de una distribución normal estándar
```

Los elementos de un vector pueden ser llamados utilizando un sub-índice. Éste inicia en 1, hasta n . Donde n es la cantidad de elementos en un vector.

```
vect1[3]
```

También es posible llamar varios elementos a la vez, si el subíndice del vector es otro vector.

```
vect1[c(3,4,5)]

vect1[3:5]
# 3:5 crea una secuencia de enteros, que incrementa en 1 a la vez.

vect1[rep(5,times=10)] # rep, es una función que repite un
```

#número un determinado número de veces.

Los vectores pueden ser datos en un archivo externo, o resultado de funciones u operaciones. A diferencia de otros lenguajes **R**, maneja vectores de una forma más intuitiva.

vec3 <- vec1 + vec2 # es un nuevo vector basado en la adición de los dos primeros.

Matrices

Las matrices son un arreglo de datos en dos dimensiones, es decir, filas y columnas. Por convención, cuando decimos que una matriz es de tamaño $f \times c$, nos referimos a que tiene f filas, y c columnas. Las filas siempre se nombran primero que las columnas.

```
(m1 <- matrix(1:9, ncol=3, byrow = T) )
```

```
(m2 <- matrix(1:9, ncol=3, byrow = F) )
```

Los elementos de una matriz se llaman por la combinación de filas y columnas a la que corresponde. Del ejemplo anterior, si quisiéramos obtener el elemento central de la matriz `m1`, lo llamamos así `m1 [2, 2]`.

Si quisiéramos llamar toda la primer columna, entonces escribimos `m1 [, 1]`. O la primer y tercer fila `m1 [c(1,3) ,]`.

Las operaciones con matrices suelen ser más delicadas y existen operadores específicos para ellas.

Marco de datos o Data frames

Esta estructura es similar a una matriz, con la diferencia, que algunas de sus columnas pueden contener *factores*, y no solo valores numéricos. Estas son las estructuras con las que representamos un diseño experimental, por ejemplo:

```
##   Var1 Var2      z
## 1   T1    a -0.506
## 2   T2    a  0.726
## 3   T1    b -2.513
## 4   T2    b  1.469
## 5   T1    c  0.677
## 6   T2    c  0.163
```

Arreglos o arrays

Estos son matrices de 3 o más dimensiones. Por ejemplo, si tenemos una serie de fotografías con la misma resolución, en el mismo lugar, podemos representar los píxeles como una matriz $f \times c$, y el tiempo como una dimensión adicional. Si ponemos en rápida sucesión las matrices, tendremos un video o película.

```
array(1:(6*2), dim=c(2,3,2))
```

Listas

Las listas son colecciones de cualquiera de los objetos anteriores (y otros que no hemos visto).

```
l1 <- list(
  vector = vect1,
  matriz = m1
)
```

Podemos llamar a los elementos de una lista, de dos formas: si conocemos el orden de los elementos de la lista, entonces, escribimos el índice dentro de dos pares de corchetes rectos: `l1[[1]]`, para llamar el vector y `l1[[2]]`, para llamar la matriz. Si conocemos los nombres de los elementos de la lista, usamos la siguiente forma: `l1$vector`, para el vector; y `l1$matriz`, para la matriz.

Una vez dentro del objeto de la lista, podemos llamar sus elementos de manera tradicional. Por ejemplo, `l1$matriz[2,2]`, para llamar el elemento central de la matriz *dentro de la lista*.

B.4. Funciones

Las funciones se representan por un nombre, seguido de un paréntesis redondo. Todo lo que esté dentro de ese paréntesis son sus argumentos. Para finalizar la función, cerramos con un paréntesis redondo derecho:

```
funcion(argumento1 = valor1, argumento2 = valor 2)
```

Nosotros podemos crear nuestras propias funciones en R. El procedimiento es sencillo:

1. Llamamos a la función con un nombre, y declaramos que se trata de una función

2. Nombramos los argumentos de la función. Podemos asignar valores por defecto. Los argumentos deben ir entre paréntesis redondos.
3. Escribimos el cuerpo de la función entre paréntesis tipo llave {}. El cuerpo de la función debe terminar con *un solo* objeto que será retornado como salida del proceso.

Por ejemplo, si queremos hacer nuestra propia función para calcular un promedio. Primero debemos entender la fórmula subyacente:

$$\bar{x} = \frac{1}{n} \sum_{i=1}^n X_i$$

Es decir, sumamos todos los elementos de un vector de valores, y lo dividimos por el *tamaño*² del vector. Entonces, nuestro argumento será un vector, y nuestra salida, un valor único con el promedio.

```
promedio <- function(vectorX){

  sumaX <- sum(vectorX)

  n <- length(vectorX) #length(), es una función que calcula
                        # el tamaño del vector.
  valor <- sumaX / n

  return(valor)

}

# Ahora probamos la función, tomando el promedio de un
# vector de números normales con media igual a cero
# Esperamos, que nuestro promedio sea un valor cercano
# a cero.

valores <- rnorm(100)

(promedio(valores))

## [1] -0.0468
```

²El tamaño del vector, se refiere al total de elementos que lo conforman

Apéndice C

Asignaciones

C.1. Tarea 01: ¡Hola mundo con *Rmarkdown*!

Objetivo: Verificar que el estudiante ha instalado, y maneja el ambiente de trabajo que se utilizará durante el curso.

Primero revisa los enlaces provistos en el [wiki](#).

Actividades

- Haz un nuevo proyecto en **RStudio**, que se llame *Tarea01*. Ver pasos en sección [B.1](#).
- En la consola de **R**, escribe `install.packages(rmarkdown)`, con todas las dependencias. O instala el paquete desde **RStudio** como se mostró en el [wiki](#).
- En **RStudio** File--> New File --> R Markdown.
- Crea una sección principal que se llame *Información profesional*.
- Luego, crea una sección secundaria que se llame *Intereses*. Usa bullets para nombrar algunos intereses profesionales.
- Luego, crea una sección secundaria llamada *Experiencia Laboral*, si aplica. Nombra algunos trabajos relacionados con el curso de Ecología de Poblaciones.
- Crea una sección principal que se llame *Integración con R*
- Consigue algunos datos interesantes en Internet. Deben ser datos para graficar, por tanto deben tener dos columnas, y varias filas. Puedes ir a [Wolfram Alpha](#). Guarda los datos como un texto delimitado por comas (. csv).

- En **R** o **RStudio** corre el comando `?read.table`. Para correr un comando en **RStudio** presiona `Cntrl + R`.
- Crea un “*chunk*” de código. Esto se hace en **RStudio**, busca un botón en la barra especial de *rmarkdown* que diga `insert`, luego escoge **R**.
- Lee la tabla y asígnala a un objeto:

```
datos <- read.table(<ruta_de_archivo_en_comillas>,  
                  header = TRUE,  
                  sep = ",")
```

- Grafica los datos en un nuevo “*chunk*”. Usa el método que prefieras. Hay mucho material de cómo hacer gráficos en **R**. Por ahora, un gráfico básico es suficiente.
- Ahora, haz otra sección llamada *Bibliografía*. En un párrafo escribe una mini-revisión de algún tema que domines y del que dispongas referencias bibliográficas. Usa los mecanismos de citas de *rmarkdown*
 - Cita en texto con `@citationKey`
 - Cita en paréntesis con `[@citationKey]`

Importante: Para que las citas funcionen, debes agregar unas opciones en la *cabecera* del documento (*YAML header*):

```
bibliography: <tu_archivo_bib>.bib  
csl: apa.csl
```

El archivo `apa.csl` se puede encontrar en google. Es un archivo de estilo APA, para dar formato a la bibliografía. Revisa [el repositorio de CSL de Zotero](#), en busca de las revistas disponibles.

-
- Por último, corre el documento con el botón `knit`. Envía el documento `.Rmd` y el `.pdf` al profesor (`dawidh15@gmail.com`).

C.2. Tarea 02: Ejercicios de crecimiento

Se recomienda hacer primero todo en papel, y luego pasarlo en limpio usando *Rmarkdown*. Uno de los aspectos más sobresalientes de LaTeX es el aspecto de las ecuaciones. Los cálculos realizados en este ejercicio son una buena excusa para investigar un poco sobre el tema. Se recomienda este [link](#)

C.2.1. Instrucciones

Resuelva el siguiente ejercicio de crecimiento exponencial

En un laboratorio se cultiva una especie presa para un programa de re-introducción de una especie de pez. En el laboratorio, se inició un proyecto de mejora en la producción de la presa, y se ha diseñado un experimento para aumentar el valor nutricional de las presas.

Se cuenta con un presupuesto de 2×10^6 CRC para la producción de animales presa en el proyecto. Además, el diseño experimental requiere de 40 recipientes acondicionados con diferentes tratamientos. Las presas crecen con una tasa de crecimiento intrínseco de $r_m = 0.098 \text{ d}^{-1}$. Además, el inóculo inicial es de 1000 individuos por recipiente. Si se sabe que el costo de mantenimiento por organismo-día es de 0.5 CRC/(ind d):

*¿Cuántos organismos por recipiente se pueden cultivar sin sobrepasar el dinero disponible?
¿Cuánto tiempo, en días, se necesitan para alcanzar esa cantidad?*

Tips:

- Este es un problema de mínimos. Primero hay que buscar la función a minimizar. Luego, uno encuentra el valor apropiado del parámetro de interés cuando la función se minimiza.
- Para minimizar una diferencia, use el valor absoluto de la diferencia: $|x - y|$.
- Use la función `optim` o `optimize`. Una vez que tenga la función que desea minimizar escrita en R use este código:

```
out <- optim(par = 0, fn = <nombre_de_funcion_para_minimizar>,
  control = list(reltol=0.01),
  method = "Brent",
  lower = <numero>,
  upper = <numero>)
```

- Antes se recomienda buscar la ayuda de la función en la consola de **R**, al escribir `?optim`. Revise los ejemplos, y lea detalladamente la ayuda.

C.3. Tarea 03: Colabora en el trabajo grupal

C.3.1. Objetivo general

Aprender la mecánica de las herramientas modernas de colaboración científica.

C.3.2. Instrucciones

Los pasos detallados para completar esta tarea se encuentran en el [Tutorial de Git y GitHub](#), y en la [Guía para hacer un Pull Request](#).

1. Haz un *Fork* del repositorio [grup2018](#).
2. Crea un nuevo proyecto en **RStudio**, usando control de versión.
3. Crea una nueva rama en **Git**, que se llame `ra1`.
4. Añade tu nombre en la sección correspondiente del archivo `informe.Rmd`.
5. Haz un commit que diga *nombre añadido*.
6. Haz un *Push* del archivo a tu repositorio.
7. Ve a **GitHub** y crea un nuevo *Pull Request*.

C.3.3. Problemas

Es posible que **Git** no te deje subir tus cambios, o hacer un *Pull Request*. Si ves un mensaje de error que diga que estás atrás con los cambios de la rama principal, y que primero debes hacer un *Pull*, significa que otro estudiante mandó sus cambios primero, y que éstos fueron incorporados por el profesor.

Entonces debes actualizarte con la última versión del `informe.Rmd`, siguiendo [estos pasos](#).

C.4. Tarea 04: Ejercicios con tablas de vida

C.4.1. Objetivo general

Valorar la importancia de las tablas de vida, para el manejo de recursos naturales.

C.4.2. Instrucciones

Lea el problema y aplique las tablas de vida para resolver la pregunta

Asuma que usted es parte de un grupo consultor que ha desarrollado un plan de manejo para un área que protege el ciclo reproductivo de una población de cabra de monte, como base alimenticia de grandes depredadores.

Después de un muestreo intensivo, se obtiene una distribución de frecuencias por edad para las hembras. Tras corregir las frecuencias en cada talla, se utiliza el método de tablas estacionarias basadas en la estructura de edad de la población, para construir las clases de edad y supervivientes. También se obtuvo un promedio de las crías hembra sobrevivientes en cada clase de edad por hembra.

Los datos obtenidos son los siguientes:

##		x	Sx	mx
##	[1,]	1	205	0.00
##	[2,]	2	96	0.00
##	[3,]	3	94	1.01
##	[4,]	4	89	1.01
##	[5,]	5	79	1.01
##	[6,]	6	68	1.01
##	[7,]	7	55	1.01
##	[8,]	8	43	1.01
##	[9,]	9	32	1.01
##	[10,]	10	22	1.01
##	[11,]	11	15	1.01
##	[12,]	12	10	1.01
##	[13,]	13	6	0.00

Los administradores del área protegida están de acuerdo, en que se considera que el plan de manejo es efectivo, si la población de cabras aumenta o se mantiene constante. Ellos le han

solicitado que usted haga esta evaluación, y les brinde los resultados.

¿Es el plan de manejo efectivo?

C.4.3. Presentación

- Presente la tarea en un archivo **R Markdown**, usando la plantilla de abajo.
- Complete las secciones del código que se solicitan.
- Corra el código y verifique que funcione, y que los resultados tienen sentido.
- De su respuesta a los administradores del área protegida.

Envíe solo el archivo `.Rmd` al profesor.

Descargue la plantilla para la tarea [aquí](#)

C.5. Tarea 05: Matrices de transición

C.5.1. Objetivo

Utilizar las matrices de transición para hacer una *Análisis de Viabilidad Poblacional* bajo diferentes escenarios de manejo.

C.5.2. Instrucciones

Resuelva el siguiente problema de gestión de recursos naturales, utilizando la teoría y práctica de matrices de transición.

C.5.3. Contexto

En las islas de Lofoten, en Escandinavia, aún existe una importante tradición ballenera. Tras años de estudio y acercamientos con los balleneros, se ha decidido permitir la caza de ballenas. Existen 15 empresas y familias balleneras y cada una clama por un permiso de caza para dos ballenas al año.

La autoridad que gestiona los recursos naturales le ha encomendado analizar el efecto de extraer las 30 ballenas al año sobre los siguientes parámetros poblacionales:

- La probabilidad de alcanzar un umbral de cuasi-extinción de 80 animales **sub-adultos y adultos** a 15 años. Para investigar el efecto de la cosecha a largo plazo.
- Evidencia gráfica de las estimaciones.

Además le piden incluir estimados de incertidumbre (excepto para la probabilidad de extinción).

El criterio de rechazo de los escenarios de manejo es el siguiente:

La chance de la probabilidad de cuasi-extinción de obtener 80 animales sub-adultos o adultos en 15 años después de la cosecha debe ser menor a 1.14

Para calcular la chance se toma la razón de la probabilidad de extinción del escenario evaluado (p_e) más uno, por la probabilidad de extinción actual (p_0) más uno:

$$o = \frac{1 + p_e}{1 + p_0}$$

Los anteriores ecólogos determinaron la matriz de transición de esta población de ballenas. Para cada parámetro de la matriz han determinado una distribución de probabilidad, y han puesto a su disposición una función para generar matrices de transición estocásticas `hacerStockMatriz`. (Esta función se encuentra en el archivo `sourceT05.R`.)

El último censo de población determinó la siguiente estructura poblacional.

$$\mathbf{n} = \begin{pmatrix} n_0 \\ n_1 \\ n_2 \end{pmatrix} = \begin{pmatrix} 153 \\ 81 \\ 43 \end{pmatrix}$$

Donde n_0 son juveniles, n_1 subadultos y n_2 adultos.

C.5.4. Instrucciones

Calcule los parámetros poblacionales solicitados distribuyendo la mortalidad de la siguiente forma:

1. Reste los treinta individuos a la clase n_0 .
2. Reste los treinta individuos a la clase n_1 .
3. Reste los treinta individuos a la clase n_2 .
4. Reste los treinta individuos equitativamente a la clase n_2 y n_1 .
5. Reste homogéneamente los treinta individuos en todas las clases.

Utilice la clase de referencia `transStochMat` (Esta función se encuentra en el archivo `sourceT05.R`). **Ver material del curso para obtener instrucciones de su uso, Matrices de Transición.**

C.6. Tarea 06

C.7. Trabajo individual

C.7.1. Objetivo general

Valorar la importancia de la estructura de trabajo del curso; al exponer al estudiante a las múltiples aplicaciones que se desarrollan en la plataforma de trabajo que se ha adoptado durante el semestre.

C.7.2. Instrucciones

El estudiante buscará un paquete en R relacionado con el tema de ecología de poblaciones o similar. No se permite el paquete **popbio**; porque ya se está utilizando en el curso.

Una vez seleccionado el paquete, el estudiante elaborará un documento **R Markdown**, con las siguientes secciones:

1. Introducción: ¿Qué hace el paquete? *Máximo 3 párrafos.*
2. Desarrolladores: ¿Quién o quienes desarrollaron el paquete, y qué experiencia tienen? *Máximo 3 párrafos*
3. Hacer un ejemplo con las funciones y datos del paquete. *Máximo 2 hojas por ambos lados.*

La mayoría de paquetes vienen con datos de prueba y tutoriales. El estudiante puede escoger una de las características principales del paquete y desarrollarla en el documento.

La idea es utilizar “*chunks*” para demostrar el uso del paquete a los compañeros.

C.7.3. Exposición

El estudiante hará una breve exposición sobre sus resultados, a partir del 28 de mayo. La exposición tendrá una duración máxima de diez minutos. El estudiante puede hacer una presentación tradicional, o una sesión en vivo del uso del paquete.

C.7.4. Sugerencias

- rMark

- Rcapture
- rDistance
- vegan

Existen muchos paquetes relacionados con el Ambiente y la Ecología en la página de [CRAN Taks View](#)

C.8. Trabajo grupal

C.8.1. Objetivo general

Aplicar los conocimientos del curso en una simulación de un proyecto colaborativo, utilizando las herramientas de trabajo vistas en el curso (**R**, **R Markdown**, **Git** y **GitHub**), para que el estudiante se familiarice con las nuevas tendencias de *Open Science*.

C.8.2. Objetivos específicos

1. Sintetizar el aprendizaje en un informe científico de la investigación, usando las herramientas de trabajo vistas en clase.
2. Valorar la importancia la justificación del procedimiento de análisis para escoger el método de toma de datos.
3. Aplicar los conocimientos teóricos del curso mediante una proyección sobre la probabilidad de extinción local de las poblaciones que vamos a muestrear.
4. Entender la importancia de verificar si nuestros resultados cumplen las suposiciones de nuestro procedimiento de análisis en la discusión del informe.

C.8.3. Justificación

La ciencia tiene principios fundamentales muy estrictos. Uno de ellos es la repetibilidad y/o reproducibilidad de los hechos con los cuales se fundamentan las hipótesis o las teorías. Los trabajos en ecología presentan problemas de repetibilidad y reproducibilidad *en el campo*, pero de hecho, sí podemos corroborar los resultados y el cumplimiento de supuestos de los análisis de otras personas. Incluso, podemos mejorar el trabajo de otros en colaboración, siempre y cuando, tengamos acceso a cierta información de estos estudios.

Muchos investigadores e instituciones, como editoriales científicas, entienden que para la ciencia es necesario un mejor flujo de información ([British Ecological Society, 2014](#); [Croucher et al., 2017](#)). El éxito de algunas iniciativas de colaboración y *Open Science*, radica en este principio. Por ejemplo, los investigadores que desarrollan análisis bioestadísticos se han beneficiado de divulgar sus procedimientos al público; ya que, investigadores similares colaboran con ellos y constantemente mejoran los análisis, corrigen errores, o sugieren nuevas alter-

nativas o métodos. Y esto solo ha ocasionado el beneficio de los numerosos usuarios de sus paquetes, y de la calidad de los análisis y por ende, las decisiones basadas en resultados.

Es importante que los estudiantes de esta licenciatura conozcan estas nuevas tendencias, y las herramientas que las han hecho posibles. Por ello, montaremos una simulación de un trabajo colaborativo en el campo de la Ecología de Poblaciones Silvestres.

C.8.4. Definiciones

mortalidad: Se refiere a la cantidad de individuos que mueren de una clase de edad o talla a la siguiente.

supervivencia: Es lo opuesto a la *mortalidad*. Es la cantidad de individuos que sobrevive de una talla a la siguiente.

fecundidad: La cantidad de huevos que produce cada individuo de *Siphonaria* spp. o de *Strombus* spp.

Matrices de transición: Es una matriz que contiene elementos de mortalidad o supervivencia y fecundidad, Se utiliza para hacer proyecciones del tamaño de una población.

fecha de cumplimiento: Es la fecha en la que una meta *ya debe haberse concretado*.

HEAD. Es un término de **Git** que se refiere a la versión más reciente de un documento que usa control de versión.

C.8.5. Metas

Meta 1, (7 %). Fecha de cumplimiento 19 de abril: Generar una revisión bibliográfica sobre temas relacionados con la ecología poblacional de *Siphonaria gigas* (o similares), y de *Strombus galeatus* (o *S. gigas*). Los estudios incluidos deben contener información sobre:

- Crecimiento: talla de los animales en relación al tiempo.
- Mortalidad: estimaciones numéricas o datos de mortalidad
- Supervivencia: estimaciones numéricas o datos de supervivencia
- Matrices de transición: todo lo relacionado con este tema.
- Análisis de viabilidad de poblaciones: todo lo relacionado con este tema.

Durante la revisión bibliográfica es importante utilizar estos puntos como criterios de aceptación para un artículo. Si no contiene información sobre estos criterios, entonces no vale la pena incluirlo en la revisión.

Si el estudiante no está familiarizado con las herramientas de búsqueda bibliográfica y bases de datos suscritas por la Universidad Nacional, es su responsabilidad informarse con la bibliotecaria de la Unidad Académica.

Meta 2, (7 %). Fecha de cumplimiento 3 de mayo: Definir el procedimiento de análisis. Se basarán en la teoría, en las investigaciones individuales, y en una búsqueda general en línea, para escoger un procedimiento de análisis que permita **calcular la probabilidad de extinción de las poblaciones de interés a cinco años**.

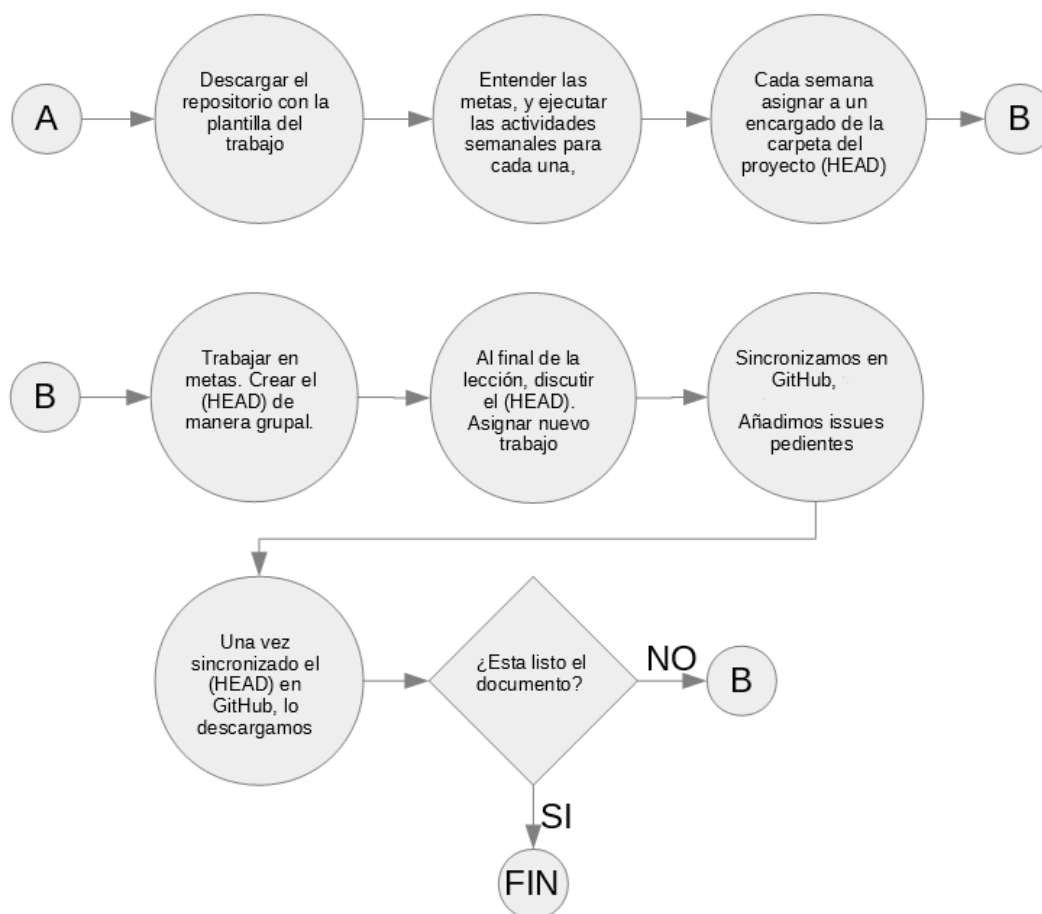
Meta 3, (5 %). Fecha de cumplimiento 10 de mayo: Diseño de muestreo. Solo hay tiempo para un sitio por población. Por tanto, el diseño de muestreo se refiere al establecimiento de las variables que se deben medir en el campo, a los formularios de campo, al procedimiento para tomar los datos y al diseño de la base de datos (se *recomienda* una base de **LibreOffice Base**).

Meta 4, (3 %). Fecha de cumplimiento 24 de mayo: Almacenamiento y control de calidad de datos. Esto se hará durante la gira, y la semana que sigue. Se debe comparar los formularios con los datos digitales en búsqueda de errores.

Meta 5, (3 %). Fecha de cumplimiento 31 de mayo: Análisis de datos. Utilizar el procedimiento de análisis y aplicarlo a los datos de campo. Idealmente, el análisis ya debería estar montado y probado en el documento principal. Por tanto, solo habría que reemplazar los datos de prueba con los reales.

Meta 6. Fecha de cumplimiento 14 de junio: Entrega del informe final. La discusión implica comparar los datos de entrada con los requisitos y supuestos del modelo, para analizar si la salida del procedimiento de análisis es válida o no. Esa fecha el informe *debe* ser entregado, para que el profesor lo revise. No se aceptan prórrogas. **El informe debe ser entregado, para hacer efectivo el porcentaje obtenido en cada una de las metas anteriores en la nota final.** Es decir, si el informe final no se entrega en esa fecha, se pierde todo el 25 % del valor.

C.8.6. Flujo semanal de trabajo



Para mantener el orden en el progreso del trabajo, se sugiere la siguiente metodología:

Primero, cada estudiante debe empezar por recopilar información sobre los géneros y especies mencionadas. Deben aplicar la lista de criterios para ver si la información es útil o no. El estudiante debería crear un archivo **R Markdown** con anotaciones sobre la información relevante del documento, e incorporar la información bibliográfica de cada documento revisado a un archivo `.bib`.

Luego, para iniciar la elaboración del informe grupal, el profesor debe haberles enseñado el uso de **Git**, y **GitHub**. En clase se explicará como crear y sincronizar una carpeta de trabajo en colaboración. El profesor será el “*editor*” del trabajo, es decir, él manejará la carpeta maestra, y cada estudiante es un colaborador.

Cada semana, los estudiantes asignarán a una persona encargada de manejar el *HEAD* (es decir, la versión más reciente del documento). Después de la sesión teórica, se discutirán

las incorporaciones y cambios del *HEAD* con respecto a la versión del *editor*. Si se aprueban los cambios en esa sesión, el estudiante enviará una propuesta de cambios al editor (en lenguaje de **Git**, esta propuesta se conoce como un *Pull Request*). El editor acepta la propuesta, y la sube al documento maestro. Posteriormente, todos los estudiantes se sincronizan con el documento maestro.

Si el editor no acepta los cambios, siempre aceptará el *HEAD* en el documento maestro, pero añadirá una serie de tareas y correcciones que los estudiantes deberán incorporar durante la semana.

Estos pasos se repiten semanalmente, hasta finalizar el documento.

Bibliografía

- Barrantes Campos, H. 2015. Introducción a las ecuaciones diferenciales. EUNED, San José, Costa Rica, 2 edición., 328 págs.
- Berryman, A. A. y P. Kindlmann. 2008. Population systems: a general introduction. Springer, 2 edición., 222 págs.
- British Ecological Society. 2014. A Guide to Data Management in Ecology and Evolution. British Ecological Society, London, UK, 1–36 págs.
- Croucher, M., L. Graham, T. James, A. Krystalli y F. Michonneau. 2017. A guide to reproducible code in ecology and evolution. British Ecological Society, London, UK, 1–42 págs.
- Elder, B. D., P. Shahani y D. F. Doak. 2003. 7 The Problems and Potential of Count-Based. *En Population Viability in Plants*, Brigham, C. y M. Schwartz, eds., tomo 165, cap. 7. Springer-Verlag, Berlin Heidelberg.
- Grafton, R. Q., J. Kirkley, T. Kompas y D. Squires. 2006. The Economics of Fishing and Fisheries Economics. *En Economics for Fisheries Management*, cap. 1. Ashgate, págs. 1–23.
- Illman, A. M., A. H. Scragg y S. W. Shales. 2000. Increase in Chlorella strains calorific values when grown in low nitrogen medium. *Enzyme and Microbial Technology* 27(8): 631–635.
- Larkin, P. 1977. An epitaph for the concept of Maximum Sustained Yield. *Transactions of the American Fisheries Society* 106(1): 1–11.
- Neal, D. 2004. Introduction to population biology. Cambridge, Cambridge, UK, 393 págs.
- Royle, J. A. y R. M. Dorazio. 2008. Hierarchical modeling and inference in ecology: the analysis of data from populations, metapopulations and communities. Academic Press, London, UK, 444 págs.
- Skalski, J. R., K. E. Ryding y J. Millspaugh. 2005. Elsevier, 656 págs.

Stubben, C. y B. Milligan. 2007. Estimating and analyzing demographic models. *Journal Of Statistical Software* 22(11): 1-23.