Probabilidades II

jmm

8 de Abril del 2015

Objetivos:

- Familiarizarse con distintas distribuciones de probabilidades y con la manera de usarlas en R.
- Simular procesos estocástcos.
- ver https://sites.google.com/site/modelosydatos/Bestiario_sp.pdf

---

R tiene funciones para todas las distribuciones de probabilidad estándares, y para cada una de estas distribuciones tenemos funciones para:

- generar valores,
- calcular probabilidades,
- probabilidades acumuladas y
- cuantiles

Estas funciones comienzan con las letras `r`, `d` ,`p` y `q` respectivamente. Por ejemplo, para la distribución de Possion tenemos: `rpois`, `dpois`, `ppois`, y `qpois`.

En muchos casos es útil poder generar muestras de una distribución en particular. Asumimos que estas muestras generadas en la computadora son una "muestra aleatoria" pero en realidad provienen de un generador de números aleatorios por lo que es más correcto decir que son "pseudo-aleatorios". Un aspecto importante, sobre todo pensando en la reproducibilidad de nuestro trabajo, es que si iniciamos al generador de números aleatorios siempre en el mismo valor, la secuencia de números pseudo-aleatorios se va a repetir y por lo tanto podemos reproducir exactamente una simulación estocástica.

En R usamos `set.seed(12345)` para inicializar el generador de números aleatorios en 12345. El número que le ponemos a `set.seed` es arbitrario.

ejemplo: simulamos una muestra de 10 valores de una distribución de Poisson

```r
set.seed(12345)
rpois(n=10,lambda=1)
```

```
##  [1] 1 2 2 2 1 0 0 1 1 4
```

Nuestros scripts pueden ser más fáciles de leer y modificar si definimos variables fuera de las funciones. Por ejemplo:

```r
n <- 100    # tamaño de la muestra
lambda <- 1.2
```

Ahora simulamos datos y vemos la frecuencia en un histograma

```r
y <- rpois(n = n, lambda = lambda)
hist(y, xlab="valores", main="")
```

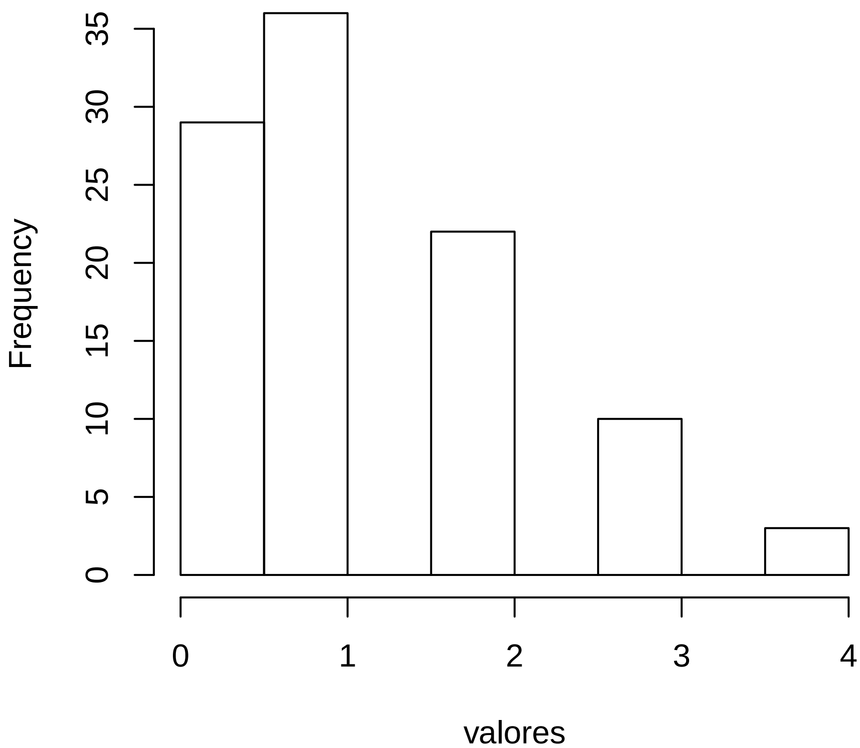

Figure 1: Histograma de datos simulados de una Poisson con $lambda = 1.2$.

Una vez que tenemos "datos" como estos, podemos ver las frecuencias relativas y compararlas con la distribución teórica usando la función `dpois`.

```r
f <- factor(y,levels=0:max(y))
obsprobs <- table(f)/n
plot(obsprobs, xlab="valores", ylab="proporción")
tprobs <- dpois(x = 0:max(y), lambda = lambda)
points(0:max(y), tprobs, pch=16, col=2)
```

Otra función útil es la Función de Distribución, que nos da la probabilidad de encontrar un valor menor o igual a un valor dado. Por ejemplo, la probabilidad de obtener $y <= 3$ con $\lambda = 1$ es:

```r
ppois(3, lambda=lambda)
```

```
## [1] 0.966231
```

Si queremos saber la probabilidad de obtener un valor *mayor* a 3 hacemos

```r
1 - ppois(3, lambda=lambda)
```

```
## [1] 0.03376897
```

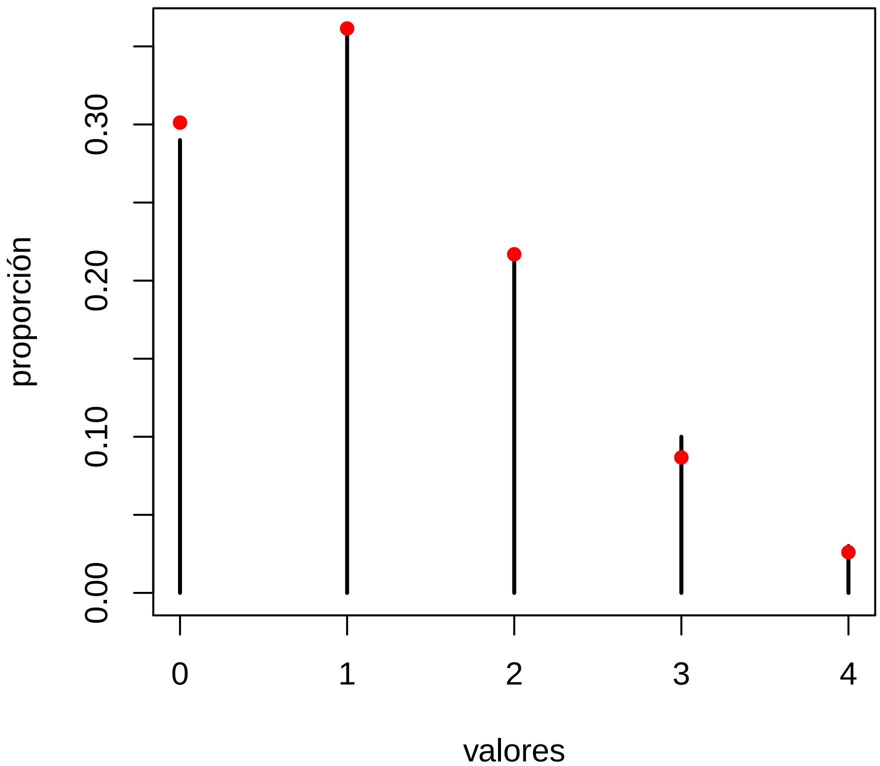

Figure 2: distribución empírica y teórica para Poisson con $lambda = 1.2$

Para ver la probabilidad de un valor en particular, por ej $y = 3$:

```
ppois(3,lambda=lambda) - ppois(2, lambda=lambda)
```

```
## [1] 0.08674393
```

¿Por qué tiene sentido hacer lo de arriba? Podemos corroborar este resultado usando `dpois`

```
dpois(3,lambda=lambda)
```

```
## [1] 0.08674393
```

El equivalente empírico de la función acumulada es la funciónn `ecdf`:

```
eF <- ecdf(y)
plot(eF, xlab="valores", ylab="Función de Probabilidad Empírica", main="")
lines( 0:6, ppois(0:6, lambda=1), type="s", col=2)
```

Por último, la función cuantil `qpois` es la inversa de la función de distribución y para un valor de probabilidad acumulada nos devuelve el valor de la variable

```
qpois(0.95, lambda=lambda)
```

```
## [1] 3
```

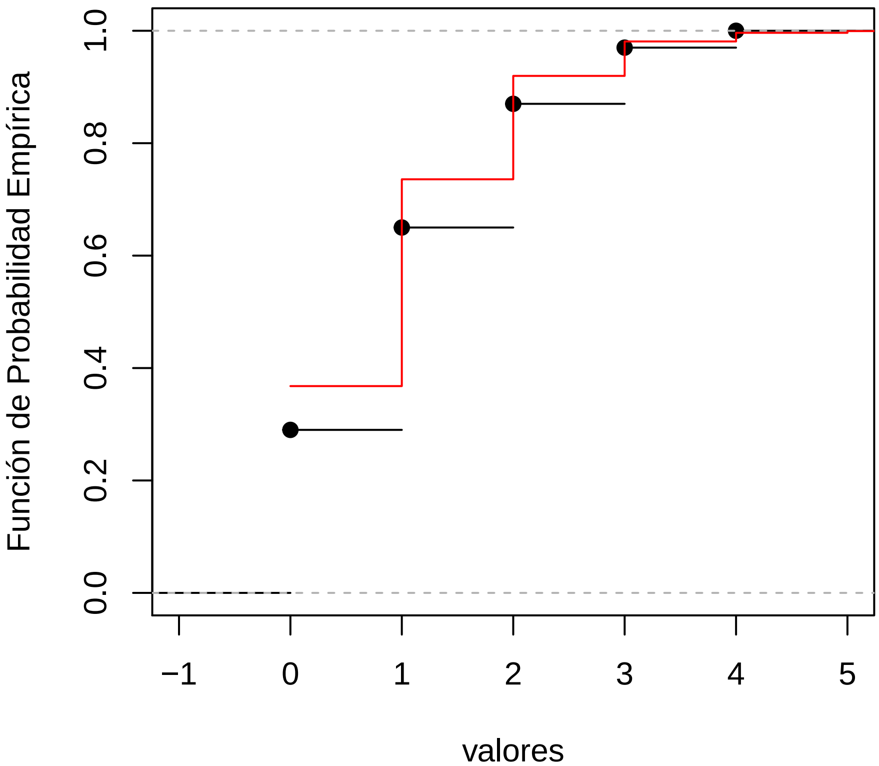

Figure 3: Acumulada empírica y teórica para Poisson con $lambda = 1.2$

La función `qpois` sirve también para calcular intervalos que contienen un porcentaje de los valores de la distribución. Por ejemplo, el 95% de los valores están entre `qpois(c(0.025,0.975) , lambda)`. El equivalente empírico es la función `quantile`

```
qpois(c(0.025,0.975) , lambda)
```

```
## [1] 0 4
```

```
quantile(y,  probs = c(0.025,0.975))
```

```
##  2.5% 97.5%
## 0.000 3.525
```

Ejercicios:

1. Asumiendo una distribución de Poisson:

- ?cu?l es la probabilidad te?rica de obtener $y = 0$ para $\lambda = 1$?
- ?cu?l es la probabilidad te?rica de $y > 2$?
- comparen la probabilidad de y=0 para $\lambda = 1$ de la distribuci?n te?rica con la obtenida emp?ricamente con muestras de 10, 100, 1000 y 10000 observaciones
- comparen las diferencias entre el valor esperado de y para la distribuci?n te?rica con los valores emp?ricos de muestras de 10, 100, 1000 y 10000 observaciones
- lo mismo para los intervalos de 95%

2. ?c?mo cambia la forma de la distribuci?n de Poisson a medida que cambia λ? (pueden ver esto simulando datos y haciendo histogramas o graficando las probabilidades te?ricas)

Simulando un Proceso Estoc?stico

Para una poblaci?n "univoltina" observamos que el n?mero de descendientes por hembra que llegan a adulto se puede representar con una distribuci?n de Possion con $\lambda = 1.2$. Suponiendo que la poblaci?n arranca en 10 individuos, para 12 generaciones podemos hacer una simulaci?n:

```
set.seed(1234)
lambda <- 1.2
niter <- 12
n <- numeric(niter)
n0 <- 10
n[1] <- n0

for(i in 2:niter){
  n[i] <- sum(rpois(n[i-1],lambda))
}

plot(n,type="o", xlab="generaciones")
```

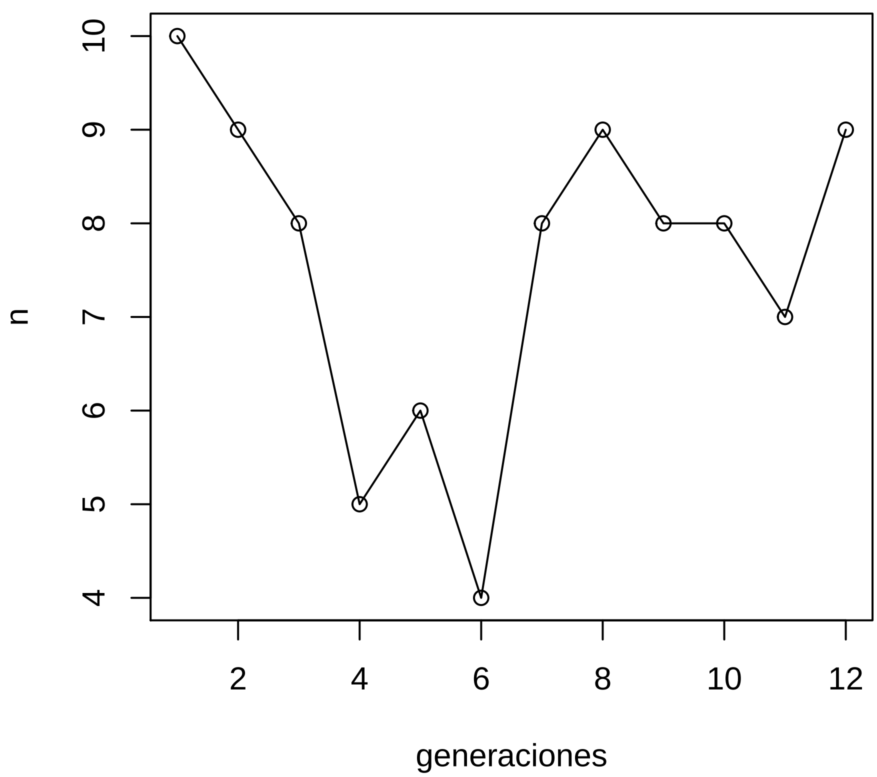

Figure 4: Una trayectoria poblacional para $n_0 = 10$ y tasa de crecimiento $\lambda = 1.2$

Pero esa es solo una posible realizaci?n de la din?mica poblacional estoc?stica. Para ver que esperamos que ocurra tenemos que hacer r?plicas:

```r
nrep <- 100
N <- matrix(NA,niter,nrep)
N[1,] <- n0
for(i in 2:niter){
  N[i,] <- rpois(nrep,lambda*N[i-1,])
}
```

```r
matplot(N, type="l", col="gray", xlab="generaciones")
lines(rowMeans(N),lwd=3)
```

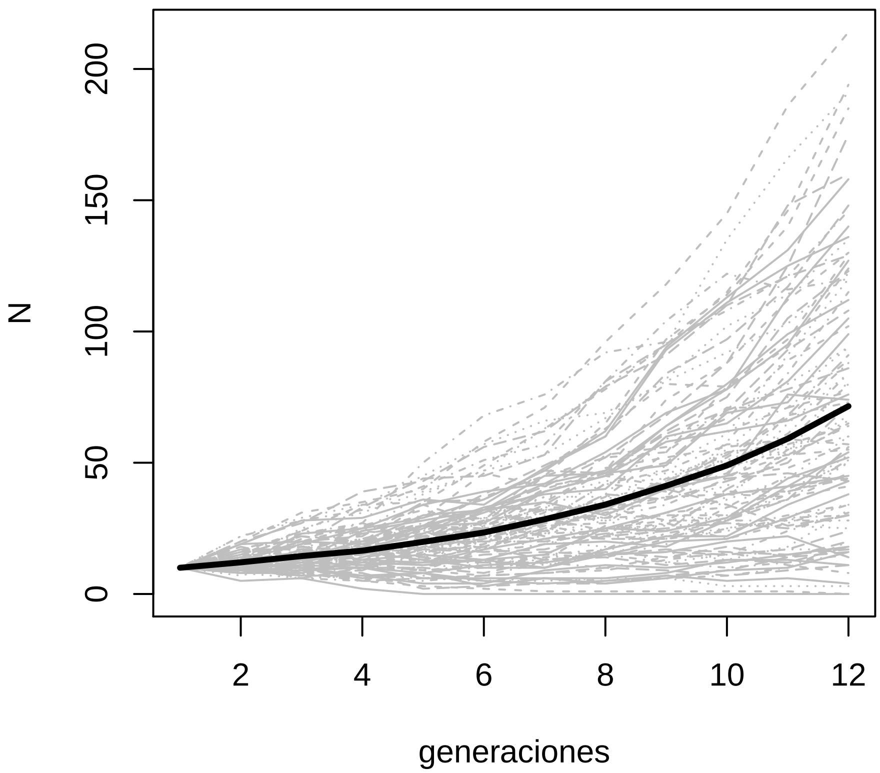

Figure 5: R?plicas de trayectorias poblacionales (en gris) y trayectoria promedio (l?nea gruesa).

Vemos que hay bastante variabilidad entre trayectorias e incluso hay casos en los que la poblaci?n se extingue:

```r
N[niter,]
```

```
##   [1]  11 194  68  80  91  45  55  34  28  24  52 214  60  88 175 127  46
##  [18]  81  34  30  76  45  73  44  43  38 102 122  20  86  14   8 191 115
##  [35] 148  74 185  66  50 160  43  57   3  89  11 105  64  32  73 108 136
##  [52] 129  53  34  18 158 124 120  31  30   4  44  94  60  72  16  44  50
##  [69]  72  56 140  41  94 120  17   0  57  65  15 129  18  64  17 146  16
##  [86] 112  77  84  51  20  54 130 135  30  65  99   0  25  84 123
```

Esta variabilidad poblacional debido a aleatoriedad en el n?mero de “reclutas” se llama **estocasticidad demogr?fica**.

Podemos incluir densodependencia haciendo que λ sea una funci?n del tama?o poblacional. Podemos hacer entonces una versi?n estoc?stica del modelo log?stico $\lambda(n) = 1 + r(1 - n(t)/K)$

```r
K <- 100
r <- 0.2
niter <- 100
nrep <- 100
N <- matrix(NA,niter,nrep)
N[1,] <- n0
for(i in 2:niter){
  N[i,] <- rpois(nrep,lambda= (1+r*(1-N[i-1,]/K))*N[i-1,])
}
```

```r
matplot(N, type="l", col="gray", xlab="generaciones")
lines(rowMeans(N),lwd=3)
```

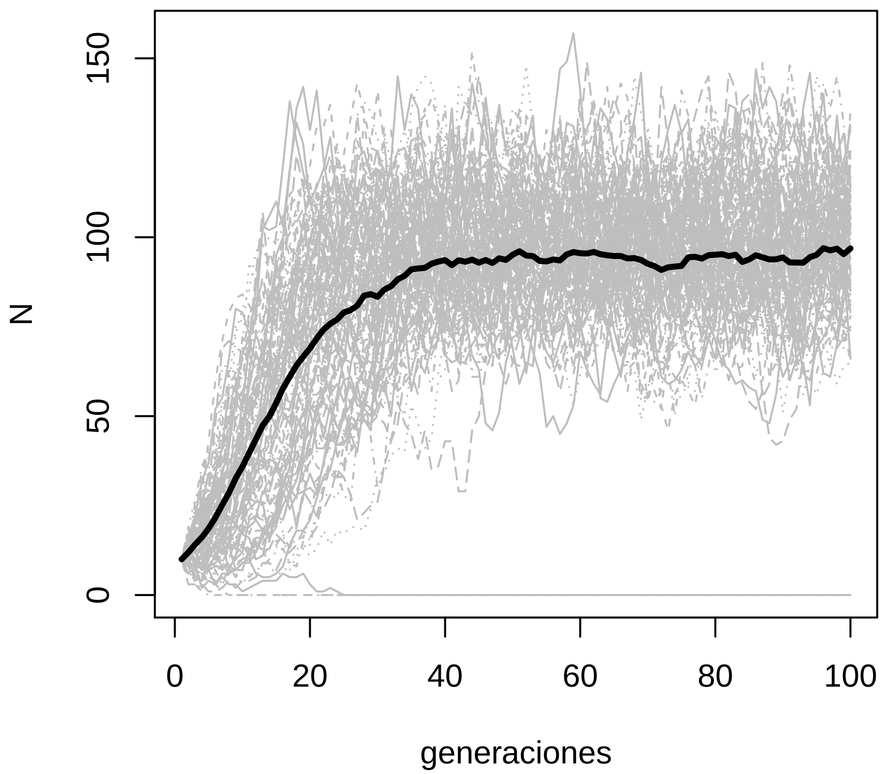

Figure 6: Modelo log?stico estoc?stico. R?plicas de trayectorias poblacionales (en gris) y trayectoria promedio (l?nea gruesa).

Hasta ahora dejamos fijos a los par?metros poblacionales, pero podr?a ser que haya a?os buenos y a?os malos, de manera que en los a?os buenos, el valor medio de supervivientes sea mayor en el modelo de crecimiento exponencial, o la capacidad de carga en el modelo log?stico sea mayor, etc.

Para el modelo de crecimiento exponencial, podemos definir $\lambda_b = 1.5$ para a?os buenos y $\lambda_m = 0.5$ para a?os malos. Si adem?s definimos que la proporci?n de a?os buenos es $p = 0.7$, tenemos que en promedio $\lambda = p\lambda_b + (1-p)\lambda_m = 1.2$ igual que antes. Sin embargo, la trayectorias poblacionales son muy diferentes. Esta variabilidad en los par?metros poblacionales se interpreta como **estocasticidad ambiental**.

Para simular a?os buenos y malos siguiendo la proporci?n p podemos generar un n?mero entre 0 y 1 con distribuci?n uniforme usando `runif` y luego ver si es mayor o menor que p. Alternativamente, podemos usar `sample` como en el ejemplo que sigue.

```r
nrep <- 3
niter <- 20
lambda_b <- 1.5
lambda_m <- 0.5

p <- 0.7
N <- matrix(NA,niter,nrep)
N[1,] <- n0
for(i in 2:niter){
  lambda <- sample(c(lambda_b,lambda_m),size=nrep,replace=TRUE,prob=c(p, 1-p))
  N[i,] <- rpois(nrep,lambda*N[i-1,])
}
```

```r
matplot(N, type="l", col=1,lwd=2, xlab="generaciones")
```

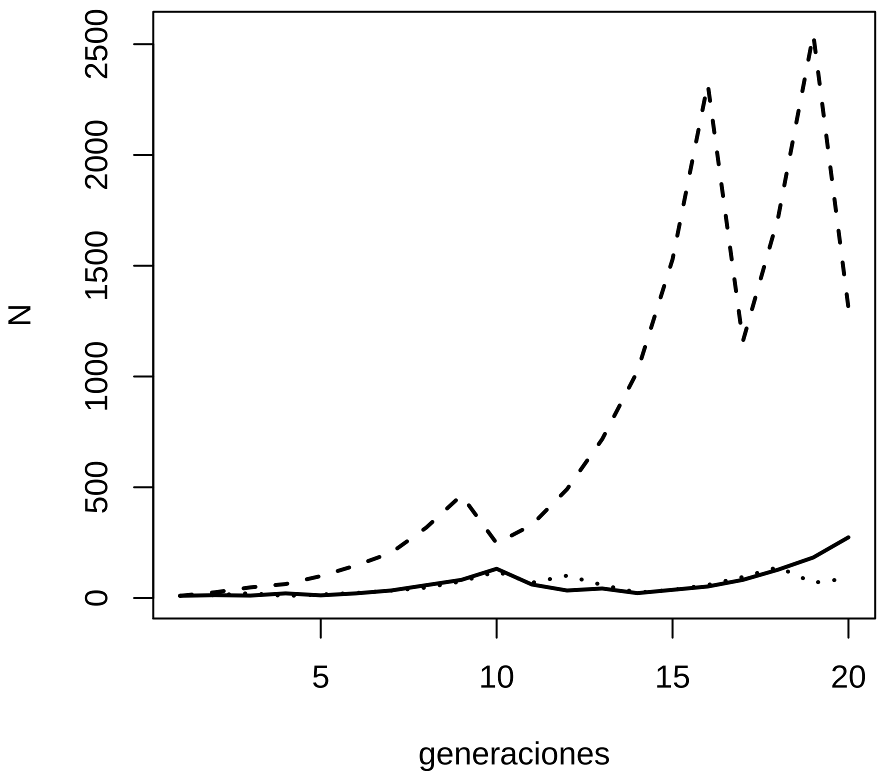

Figure 7: R?plicas de trayectorias poblacionales con etocasticidad ambiental.

Desaf?o:

Armar una simulaci?n para estocasticidad ambiental con autocorrelaci?n temporal y ver c?mo cambia la probabilidad de extinci?n en 30 generaciones seg?n el grado de autocorrelaci?n. Usar los mismos par?metros que en el ejemplo de arriba