

Ejercicios Bucles y Condiciones I

EJERCICIOS

Ejercicio 1: Fotosíntesis con Variación de Luz

Un equipo de botánicos está estudiando la producción de glucosa mediante fotosíntesis en una especie de planta mediterránea. Han observado que la intensidad de la luz solar varía según la función:

$L(t) = L * (1 + \sin(t/3) * 0.2)$, donde t es el número de ciclo.

Se requiere modelar la producción de glucosa bajo estas condiciones durante 20 ciclos o hasta que la producción acumulada de glucosa alcance 1000 unidades. La producción de glucosa viene definida por:

$P_glucosa \leftarrow P[t] + E * L - D * P[t]$

Implemente este modelo, mostrando la intensidad de luz y la producción de glucosa en cada ciclo, así como el momento en que se alcanza la producción máxima, si ocurre.

DATOS:

Producción inicial de glucosa es 0

Intensidad de luz inicial es 100

Eficiencia fotosintética: $E \leftarrow 0$

Tasa de degradación de glucosa: $D \leftarrow 0.1$

Ejercicio 2: Crecimiento Poblacional con Factores Ambientales

Un biotecnólogo está estudiando el crecimiento de una población de conejos en un ecosistema. El modelo de crecimiento considera factores ambientales que varían con el tiempo, como la disponibilidad de alimentos y la presencia de depredadores. La población en el tiempo t se describe mediante la siguiente ecuación:

$$N_0 \cdot \prod_{i=1}^t (1 + r_i) + \sum_{i=1}^t F_i$$

Donde:

- $N(t)$ es la población en el tiempo t
- N_0 es la población inicial (100 conejos)
- r_i es la tasa de crecimiento en el intervalo i (varía entre 0.1 y 0.3)
- F_i es el factor ambiental que afecta a la población en el intervalo i (puede ser positivo o negativo)

Dados los siguientes valores para 5 intervalos de tiempo:

- $r = [0.2, 0.15, 0.25, 0.1, 0.3]$
- $F = [-10, 5, -8, 15, -5]$

Calcule la población de conejos para cada intervalo hasta $t=5$. Si en algún momento la población cae por debajo de 50, el cálculo debe detenerse.

Ejercicio 3: Eficiencia Metabólica en Cadena de Reacciones

Un bioquímico está estudiando una cadena de reacciones metabólicas en una bacteria. La eficiencia total del proceso se modela mediante la siguiente ecuación:

$$E = \sum_{i=1}^n k_i \cdot \prod_{j=1}^i e_j$$

Donde:

- E es la eficiencia total del proceso
- n es el número total de reacciones en la cadena (5)
- k_i es el coeficiente de importancia de la reacción i
- e_j es la eficiencia individual de la reacción j

Dados los siguientes valores:

- $k = [0.2, 0.15, 0.3, 0.25, 0.1]$
- $e = [0.9, 0.85, 0.95, 0.8, 0.9]$

Calcule la eficiencia total del proceso. Si en algún punto la eficiencia acumulada cae por debajo de 0.6, el cálculo debe detenerse y reportar la eficiencia hasta ese punto.

Ejercicio 4: Respuesta Inmune Acumulativa

Un inmunólogo está modelando la respuesta inmune de un organismo a una serie de exposiciones a un antígeno. La respuesta inmune total después de n exposiciones se define como:

$$R(n) = R_0 + \sum_{i=1}^n (A_i \cdot \prod_{j=1}^i (1 + M_j))$$

Donde:

- $R(n)$ es la respuesta inmune total después de n exposiciones
- R_0 es la respuesta inmune basal (10 unidades)
- A_i es la intensidad de la i -ésima exposición al antígeno
- M_j es el factor de memoria inmunológica para la j -ésima exposición

Dados los siguientes valores para 5 exposiciones:

- $A = [5, 8, 12, 7, 10]$
- $M = [0.1, 0.15, 0.2, 0.25, 0.3]$

Calcule la respuesta inmune total para cada exposición hasta $n=5$. Si en algún momento la respuesta inmune supera 100 unidades, el cálculo debe detenerse y reportar ese valor como la respuesta máxima.

Ejercicio 5: Producción de Proteínas con Regulación por Retroalimentación

Un biotecnólogo está estudiando la producción de una proteína específica en una célula. La cantidad de proteína producida en cada ciclo se modela según la siguiente ecuación:

$$P(t) = P_0 + \sum_{i=1}^t (E_i \cdot \prod_{j=1}^i (1 + A_j)) - \sum_{i=1}^t D_i$$

Donde:

- $P(t)$ es la cantidad de proteína en el tiempo t
- P_0 es la cantidad inicial de proteína (50 unidades)
- E_i es la eficiencia de producción en el ciclo i
- A_j es el factor de disponibilidad de aminoácidos en el ciclo j
- D_i es la tasa de degradación de la proteína en el ciclo i

Además:

- Si $P(t)$ excede un umbral superior de 200 unidades, se aplica un coeficiente inhibitor de 0.7 a la producción en el siguiente ciclo.
- Si $P(t)$ cae por debajo de un umbral inferior de 30 unidades, se aplica un coeficiente estimulador de 1.5 a la producción en el siguiente ciclo.

Dados los siguientes valores para 6 ciclos:

- $E = [10, 12, 15, 11, 13, 14]$
- $A = [0.05, 0.08, 0.06, 0.07, 0.09, 0.08]$
- $D = [5, 6, 7, 5, 8, 6]$

Calcule la cantidad de proteína para cada ciclo hasta $t=6$. El cálculo debe detenerse si la cantidad de proteína supera 300 unidades o cae por debajo de 10 unidades.

SOLUCIONES

Ejercicio 1: Fotosíntesis con Variación de Luz

```
cat("\014")
rm(list=ls())

P <- 0 # Producción inicial de glucosa
L <- 100 # Intensidad de luz inicial
E <- 0.5 # Eficiencia fotosintética
D <- 0.1 # Tasa de degradación de glucosa
iteracion_max <- 20

for (t in 1:iteracion_max) {
  L <- L * (1 + sin(t/3) * 0.2) # Variación sinusoidal de la luz

  P_glucosa <- P[t] + E * L - D * P[t]

  if (P_glucosa > 1000) {
    cat("Producción máxima alcanzada en t=", t, "\n")
    break
  }

  P <- c(P, P_glucosa)

  cat("t=", t, ": Producción de glucosa =", P_glucosa, ", Cantidad de luz=",
L, "\n")
}
```

Ejercicio 2: Crecimiento Poblacional con Factores Ambientales

```
cat("\014")
rm(list=ls())

N0 <- 100 # Población inicial
r <- c(0.2, 0.15, 0.25, 0.1, 0.3) # Tasas de crecimiento
F <- c(-10, 5, -8, 15, -5) # Factores ambientales
N <- c(N0) # Vector para almacenar los resultados

for (t in 1:5) {
  producto <- N0 * prod(1 + r[1:t])
  suma <- sum(F[1:t])
  Nt <- producto + suma

  if (Nt < 50) {
    cat("La población cayó por debajo de 50 en t =", t, "\n")
    break
  }

  N <- c(N, Nt)
  cat("t =", t, ": Población de conejos es: ", Nt, "\n")
}
```

Ejercicio 3: Eficiencia Metabólica en Cadena de Reacciones

```
cat("\014")

rm(list=ls())

k <- c(0.2, 0.15, 0.3, 0.25, 0.1) # Coeficientes de importancia
e <- c(0.9, 0.85, 0.95, 0.8, 0.9) # Eficiencias individuales
E <- 0 # Eficiencia total

for (i in 1:5) {
  producto <- prod(e[1:i])
  if (producto < 0.6) {
    cat("La eficiencia acumulada cayó por debajo de 0.6 en i =", i, "\n")
    break
  }

  E <- E + k[i] * producto

  cat("Iteración =", i, ": Eficiencia acumulada =", producto, ", Eficiencia
total =", E, "\n")
}
```

Ejercicio 4: Respuesta Inmune Acumulativa

```
cat("\014")
```

```
rm(list=ls())
```

```
R0 <- 10 # Respuesta inmune basal
```

```
A <- c(5, 8, 12, 7, 10) # Intensidades de exposición
```

```
M <- c(0.1, 0.15, 0.2, 0.25, 0.3) # Factores de memoria inmunológica
```

```
R <- c(R0) # Vector para almacenar los resultados
```

```
for (n in 1:5) {
```

```
  suma <- sum(A[1:n] * cumprod(1 + M[1:n]))
```

```
  # cumprod() en R calcula los productos acumulados de las componentes  
  # de un vector. También se podría haber desarrollado un for manualmente
```

```
  Rn <- R0 + suma
```

```
  if (Rn > 100) {
```

```
    cat("La respuesta inmune superó 100 unidades en n =", n, "\n")
```

```
    break
```

```
  }
```

```
  R <- c(R, Rn)
```

```
  cat("n =", n, ": Respuesta inmune =", Rn, "\n")
```

```
}
```

```
cat("Respuesta total =", Rn, "\n")
```

Ejercicio 5: Producción de Proteínas con Regulación por Retroalimentación

```
cat("\014")

rm(list=ls())

P0 <- 50 # Cantidad inicial de proteína

E <- c(10, 12, 15, 11, 13, 14) # Eficiencia de producción

A <- c(0.05, 0.08, 0.06, 0.07, 0.09, 0.08) # Factor de disponibilidad de aminoácidos

D <- c(5, 6, 7, 5, 8, 6) # Tasa de degradación

P <- c(P0) # Vector para almacenar los resultados

umbral_superior <- 200

umbral_inferior <- 30

coef_inhibidor <- 0.7

coef_estimulador <- 1.5

coef_actual <- 1

for (t in 1:6) {

  producto <- sum(E[1:t] * cumprod(1 + A[1:t])) * coef_actual

  suma <- sum(D[1:t])

  Pt <- P0 + producto - suma

  if (Pt > 300) {

    cat("La cantidad de proteína superó 300 unidades en t =", t, "\n")

    break

  } else if (Pt < 10) {
```

```
cat("La cantidad de proteína cayó por debajo de 10 unidades en t =", t,
"\n")
```

```
break
```

```
}
```

```
P <- c(P, Pt)
```

```
cat("t =", t, ": Cantidad de proteína =", Pt, "\n")
```

```
if (Pt > umbral_superior) {
```

```
  coef_actual <- coef_inhibidor
```

```
  cat("Se aplicará un coeficiente inhibidor en el siguiente ciclo\n")
```

```
} else if (Pt < umbral_inferior) {
```

```
  coef_actual <- coef_estimulador
```

```
  cat("Se aplicará un coeficiente estimulador en el siguiente ciclo\n")
```

```
} else {
```

```
  coef_actual <- 1
```

```
}
```

```
}
```