

LABORATORIO DE PROCESAMIENTO DE INFORMACION METEOROLÓGICA

Lenguaje de Programación R

EJERCICIOS

entre [] me devuelve la posición

`x<-c(2,5,3,7,1,8,9,4,6)`

`sum(x)` # suma todos los elementos

`length(x)` # largo total

`sum(x<5)` #cuantos cumplen la condición `x<5`, vector lógico

`x[x<5]` # quienes cumplen la condición

`sum(x[x<5])` # ahora si, los sumo

`length(x[x<5])` cantidad de elementos x que suma para llegar a ese valor que sumé

`x*(x<5)` # el valor de x solo en las posiciones TRUE producto entre el valor de x y 1(por el verdadero) y cero por el falso. Creo que Me da la misma cantidad de elementos que tenía

`sum(x*(x<5))` # ahora los sumo suma los elementos que son distintos de cero

`which(x<5)` # que posiciones cumplen con la condición which me devuelve el resultado de lo que le pregunto. Acá posiciones donde `x<5`. Si quiero los falsos tengo que cambiar la condición. Which guarda posiciones

`sum(which(x<5))` # sumo posiciones, NO valores

`z<-numeric(10)` # genero otro vector vector de 10 posiciones pero no le asigna valores

`id<-which(x<5)` # en "id" asigno los valores que cumplen la condición guarda las posiciones que cumplen `x<5`

`z[id] <- x[x<5]`

`z[which(x<5)] <- x[x<5]` # equivalente al anterior (¿mejor?)

EJERCICIOS PARA QUE PIENSEN

`x<-c(2,1,0,3,6,1)`

1) ¿Qué ocurre si hacemos `x[1:3]` cuando `x` tiene los datos 2; 1; 0; 3; 6; 1? le pido el valor en esas posiciones

2) `1:5` # genera los valores 1 2 3 4 5 genera la secuencia del 1 al 5
 `2*1:5` # ¿qué operador “gana”? `2*secuencia` genera un vector de 5 elementos
 multiplica cada valor por 2
 genera la secuencia y lo multiplica por 2

3) ¿Pueden predecir el valor de la expresión `1:7*1:2`? multiplica secuencias
 lo hace pero pone error porque no son múltiplos

4) ¿Qué pasa con esta expresión `1:8*1:2`?

EJEMPLO DE RESOLUCION DE PROBLEMA

De un edificio, a una altura de 15 m, se ha lanzado con un ángulo de 50 grados, un proyectil a una velocidad de 7 m/s.

¿Cuáles serán las alturas (coordenadas y) del proyectil a cada 0.5 m de distancia horizontal desde donde se lanzó y hasta los 11 m?

Ecuaciones:

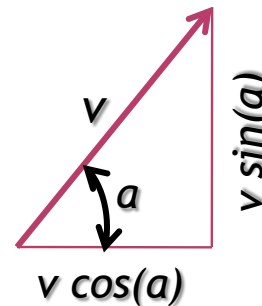
conozco valores de x (cada 0,5), velocidad inicial y g
desconozco valores de t y de y

$$x = v_{0x} * t + x_0 \quad t = x/v_0 \text{ (despejo t asumiendo que } x_0 \text{ es cero)}$$
$$y = -1/2 g * t^2 + v_{0y} * t + y_0 \quad y \text{ es un vector de 22 elementos}$$

g : aceleración de la gravedad, t : tiempo, v_{0x} y v_{0y} : componentes de la velocidad.

La Figura muestra como obtener a partir de la velocidad inicial y el ángulo, las componentes de velocidad.

En R, los argumentos de las funciones trigonométricas deben estar dados en *radianes*



EJEMPLO (cont)

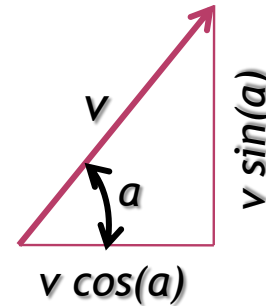
De un edificio, a una altura de 15 m, se ha lanzado con un ángulo de 50 grados, un proyectil a una velocidad de 7 m/s.

¿Cuáles serán las alturas (coordenadas y) del proyectil a cada 0.5 m de distancia horizontal desde donde se lanzó y hasta los 11 m?

Definimos en R nuestra variables iniciales:

Cuales son????

```
g <- 9.81           # aceleración gravedad
x0 <- 0              # x inicial
y0 <- 15             # y inicial
vi <- 7              # velocidad inicial
alphaD <- 50         # ángulo-grados
```



Y las componentes de la velocidad son....

```
alpha <- (pi/180) * alphaD           # ángulo-radianes porque los ángulos tienen que ir en radianes
vox <- vi * cos(alpha)                # componente x de velocidad inicial
voy <- vi * sin(alpha)                # componente y de velocidad inicial
vox; voy
[1] 4.499513
[1] 5.362311
```

EJEMPLO (cont)

De un edificio, a una altura de 15 m, se ha lanzado con un ángulo de 50 grados, un proyectil a una velocidad de 7 m/s.

¿Cuáles serán las alturas (coordenadas y) del proyectil a cada 0.5 m de distancia horizontal desde donde se lanzó y hasta los 11 m?

Ahora obtenemos las x para las que se desea hacer el cálculo

Cuales son????

desde 0 hasta 11 de 0.5 en 0.5 *Como lo genero???*

```
las.x <- seq(from = 0, to = 11, by = 0.5)
```

Como se relacionan x e y???

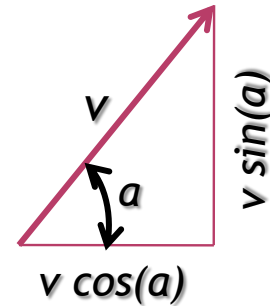
$$x = v_{0x} * t + x_0$$
$$y = -1/2 g * t^2 + v_{0y} * t + y_0$$

Para cada x obtengo el t y para ese tiempo obtengo la altura (y) a la que se encuentra el proyectil

$$t = (x - x_0) / v_{0x}$$

```
las.t <- (las.x - x0) / vox
```

```
las.y <- -(g/2) * las.t^2 + voy * las.t + y0
```



EJEMPLO (cont)

De un edificio, a una altura de 15 m, se ha lanzado con un ángulo de 50 grados, un proyectil a una velocidad de 7 m/s.

¿Cuáles serán las alturas (coordenadas y) del proyectil a cada 0.5 m de distancia horizontal desde donde se lanzó y hasta los 11 m?

Los resultados:

las.x

[1] 0.0 0.5 1.0 1.5 2.0 2.5 3.0 3.5 4.0 4.5 5.0

[12] 5.5 6.0 6.5 7.0 7.5 8.0 8.5 9.0 9.5 10.0 10.5

[23] 11.0

las.y

[1] 15.0000 15.5353 15.9495 16.2425 16.4144 16.4652 16.3948

[8] 16.2033 15.8906 15.4568 14.9019 14.2258 13.4286 12.5103

[15] 11.4708 10.3102 9.0285 7.6256 6.1015 4.4564 2.6901

[22] 0.8026 -1.2059

INTRODUCCIÓN A LA LÓGICA DE LA PROGRAMACIÓN

- Expresiones lógicas
- Operadores lógicos
- Operadores relacionales
- Diagramas de Flujo
- Estructuras de control repetitivas (Funciones iterativas FOR, WHILE, REPEAT)
- Estructura condicional simple (Función IF - ELSE - IFELSE, SWITCH)
- Interrupciones de ciclos (BREAK, NEXT, RETURN)

EXPRESIONES LÓGICAS

Las expresiones lógicas proporcionan respuestas del tipo Verdadero/Falso.

- o Los test lógicos se basan en operadores relacionales
- o Los operadores relacionales se combinan entre si mediante operadores lógicos

Operadores de Relación

>	• Mayor
>=	• Mayor o igual
<	• Menor
<=	• Menor o igual
==	• Igual
!=	• Distinto

porque con un = solo estaría asignando
!= es la negación

Ejemplo

Combinamos con operadores lógicos

$x > 3 \text{ \& } y > 3$
[1] TRUE

$x > 3 \text{ \& } y < 3$
[1] FALSE

Ejemplo

Definimos 2 variables

$x = \pi$
 $y = 2 * \pi$

Interrogamos con variables de relación

$x \geq 3$
[1] TRUE

$x < 3$
[1] FALSE

Operadores Lógicos

&	• Intersección (and/y)	cumplen las dos cosas
	• Unión (or/o)	
!	• Negación	

EXPRESIONES LÓGICAS

Las expresiones lógicas pueden utilizarse en expresiones aritméticas, en cuyo caso se transforman primero en expresiones numéricas

FALSE \rightarrow 0

TRUE \rightarrow 1

Hay casos en que un vector lógico y el numérico no son equivalentes.

`x == NA`

[1] NA

`x == NaN`

[1] NA

no está bien el resultado porque me debería dar un vector lógico y me está diciendo que es NA
NA y NaN no me dan resultados lógicos. Voy a tener que usar `is.na(x)`

NA no es realmente un valor, sino un indicador de una cantidad que no está disponible
La función **`is.na(x)`** crea un vector lógico del tamaño de **`x`** cuyos elementos solo valdrán:
TRUE si el elemento correspondiente de **`x`** es NA
FALSE en caso contrario

`is.na(x)`

[1] FALSE

`is.na(x)` es TRUE tanto para los valores NA como para los NaN
Para diferenciar existe la función **`is.nan(x)`** que solo toma el valor TRUE para valores NaN

`x == Inf`

[1] FALSE

devuelve un resultado lógico porque matemáticamente infinito es un número

`is.infinite(x)`

[1] FALSE

OPERADORES LÓGICOS

La aritmética entre objetos lógicos se puede llevar a cabo con los siguientes operadores lógicos:

Operador	Operación
<code>!x</code>	Negación de x. Los T los convierte en F y viceversa.
<code>x&y</code>	Intersección, operador lógico y: T y T da T, otra comparación da F.
<code>x y</code>	Unión, operador lógico o: F y F da F, otra comparación da T.
<code>xor(x,y)</code>	Exclusivo OR, xor (T,F)==T, otra comparación da F.
<code>all</code>	Para una secuencia de argumentos lógicos, all devuelve el valor lógico que indica si todos los elementos son TRUE.
<code>any</code>	Para una secuencia de argumentos lógicos, any devuelve el valor lógico que indica si algún elemento es TRUE.

EJEMPLO

```
x<-c(T,T,F,F)
```

x e y son vectores lógicos acá, no numéricos

```
y<-c(T,F,T,F)
```

```
x&y # produce TRUE FALSE FALSE FALSE
```

ambos deben ser verdaderos

```
x|y # produce TRUE TRUE TRUE FALSE
```

uno de los dos tiene que ser verdadero

```
xor(x,y) # produce FALSE TRUE TRUE FALSE
```

uno solo verdadero, NO los dos

```
any(x) # produce TRUE
```

si cualquiera de las x son Verdaderas. con que haya una, ya me pone True

```
all(x) # produce FALSE
```

si TODAS las x son verdaderas

EXPRESIONES LÓGICAS (CONT)

Otro ejemplo de aplicación, asignando resultados lógicos a una variable

`rm(list=ls())` Borra todas las variables

`seq(0, 3*pi, by=.1) -> x` Genera un vector con valores entre 0 y 3π cada 0.1.
Lo guarda en x

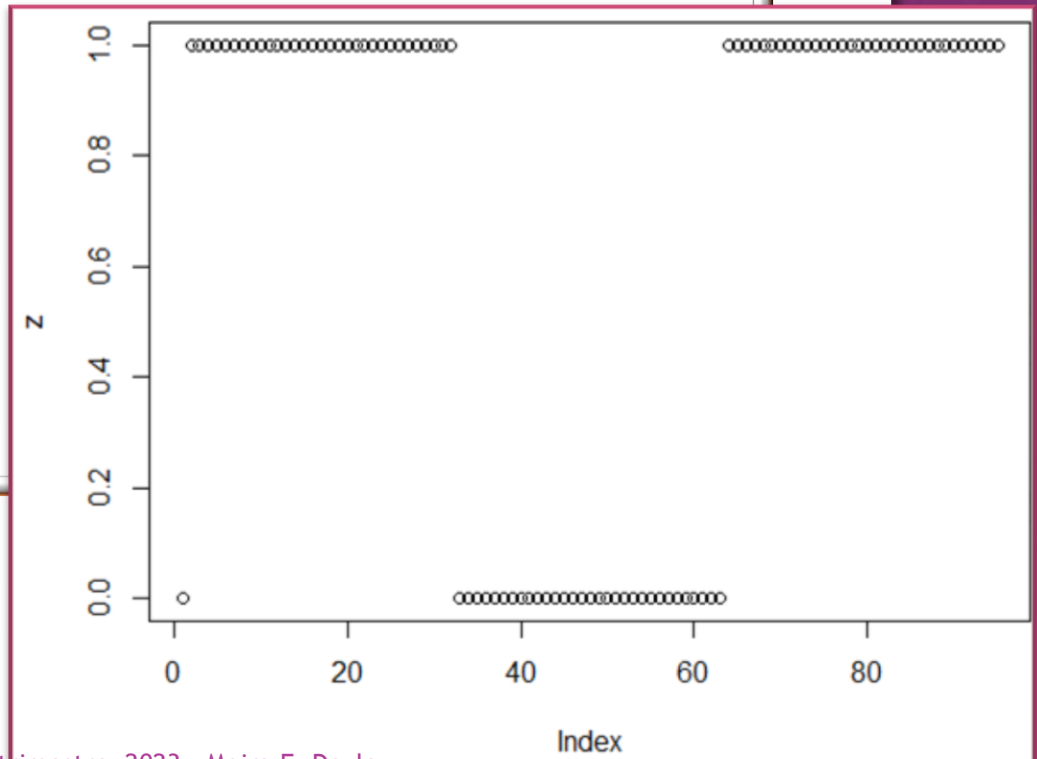
`y=sin(x)`

`z=(y>0)` interrogación lógica... si $y>0$ **TRUE** sino **FALSE**

`z`

```
[1] FALSE TRUE TRUE TRUE TRUE TRUE TRUE TRUE
[9] TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE
[17] TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE
[25] TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE
[33] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
[41] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
[49] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
[57] FALSE FALSE FALSE FALSE FALSE FALSE FALSE TRUE
[65] TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE
[73] TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE
[81] TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE
[89] TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE
```

`plot(z)`



esquemas

DIAGRAMAS DE FLUJO

me sirve para ordenar la secuencia de pasos

Es un diagrama que describe un proceso, sistema o algoritmo informático. Se usan ampliamente en numerosos campos para documentar, estudiar, planificar, mejorar y comunicar procesos que suelen ser complejos en diagramas claros y fáciles de comprender

DIAGRAMA DE FLUJO

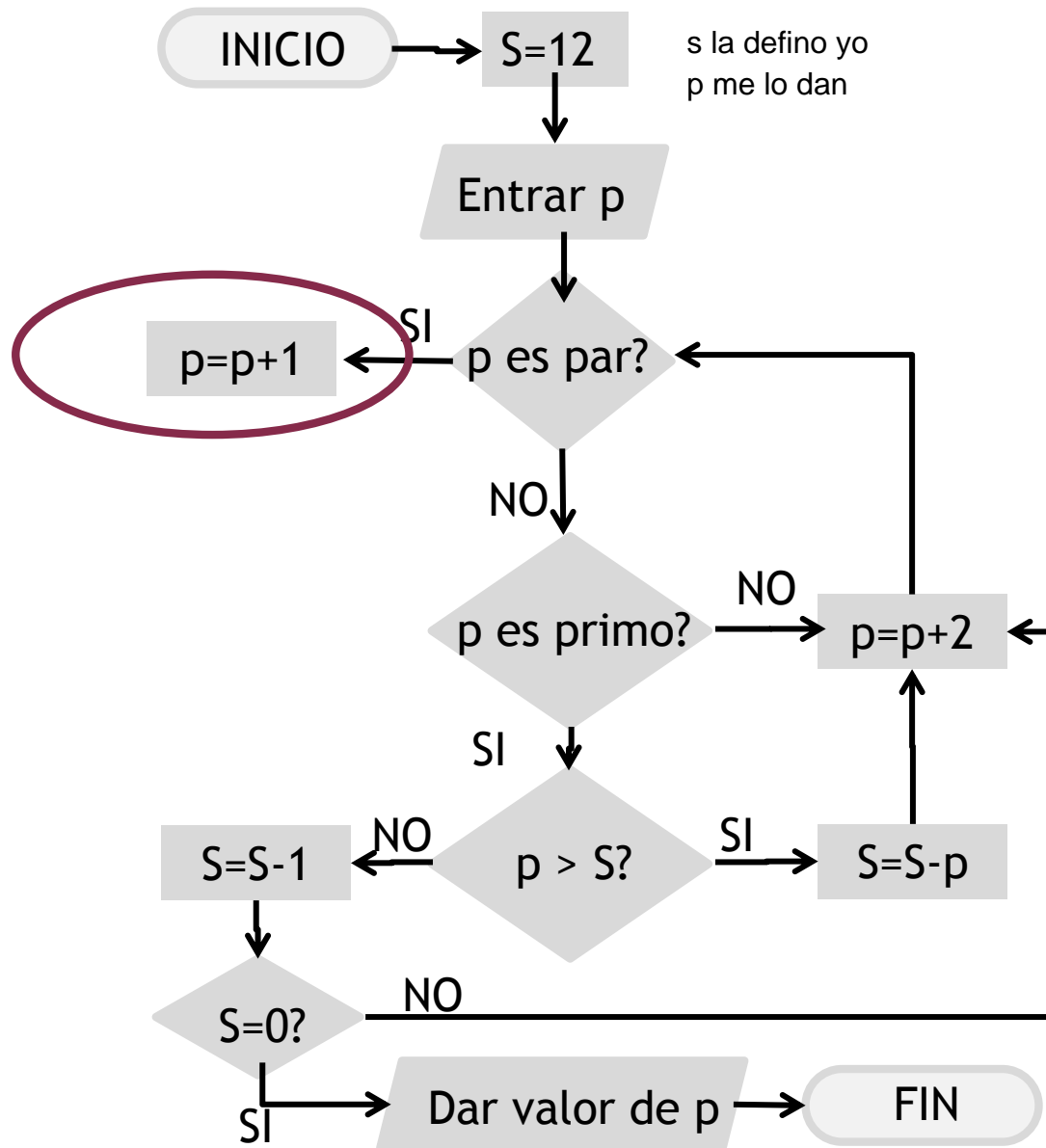


DIAGRAMA DE FLUJO



Terminal / Terminador principio y fin



Proceso



Datos



Decisión pregunta



Flecha de Flujo conectan los distintos procesos



Conector

NOS EJERCITAMOS....

Realizar el diagrama de flujo de los siguientes problemas

1. Preparar una ensalada con tres ingredientes
2. Llegar a mi destino cuando salgo de Ciudad Universitaria
3. Restar dos números ingresados por el usuario de tal modo que el resultado sea positivo. Considerar la posibilidad de que el usuario ingrese valores no numéricos o mas datos de los pedidos.