


Contenidos clase 1 de : variables, vectores, vectores aleatorios, matrices, arrays, y listas.

Variables

1. ¿Qué pasa si escribimos en línea de comandos $b < -3$? ¿Por qué una flecha ($<-$) y por qué apunta hacia b ?
2. ¿Hay alguna diferencia entre $b = 3$, $b < -3$ y $3 > b$? ¿Qué asignación le parece más elegante (genera menos confusión)? ¿Por qué? *La flecha es una manera de asignar que se presta a menos confusión.*
3. A partir de la variable b creada en el ítem anterior genere una variable d que sea el doble de b .
4. Sume b con d , multiplique b con d , divida b sobre d . ¿El resultado quedó guardado en algún lado (alguna variable)? ‘
5. ¿Se puede usar  como una calculadora sin definir variables? *Sí, lo acabamos de hacer...*
6. ¿Qué pasa si escribimos $b < -hola$? , ¿y si escribimos $b < -"hola"$?

Vectores

Vectores de números:

7. Resolver
 - (a) Defina el vector x que tiene coordenadas $(1, 4)$.
 - (b) Defina el vector y que tiene coordenadas $(6, 9, 12, 412)$.
 - (c) Defina el vector z que tiene coordenadas $(1, 2, \dots, 1223)$.
 - (d) Defina el vector w que tiene coordenadas $(0.1, 0.2, 0.3, \dots, 422.7)$. **Hint:** una opción es usar `seq(★1, ★2, ★3)`
 - (e) ¿Qué longitud tiene el vector w ? **Hint:** use la función `length`
 - (f) ¿Cuál es el valor de w en la coordenada 154?
 - (g) Reemplace el valor que se encuentra en la coordenada 154 del vector w por el valor 2.4.
 - (h) Diga que valores toma w entre las coordenadas 124 y la 162. **Hint:** ¿Qué pasa si escribe `w[2 : 6]`?
 - (i) Defina un nuevo vector s que tenga primero las coordenadas de x y luego las de w . **Hint:** se pueden concatenar vectores.

-
- (j) Defina un vector t que sea igual a w salvo que no tiene la coordenada 124. **Hint:** ¿En qué difiere $y[-2]$ de y ?
- (k) Utilizando permutaciones de coordenadas construya un vector s que sea igual a y salvo que las coordenadas 2 y 4 están intercambiados. **Hint:** ¿Qué pasa si escribe $y[c(2,1,3,4)]$?
8. Defina el vector y que tiene coordenadas (6, 9, 12, 16) utilizando el comando $y <- scan()$. **Hint:** Para salir del scan hay que darle enter.
9. Resolver:
- (a) Defina un vector x con los números naturales del 1 al 10, y un vector y con los números del 101 al 110. ¿Qué se obtiene si hace $x + y$? ¿Qué se obtiene si hace x/y ? ¿Es importante que los vectores tengan la misma longitud? Grafique y en función de x utilizando el comando `plot`
¿Qué hace `R` si tiene que sumar vectores de distinta longitud? **Hint:** Si le exigimos que sume vectores de distinta dimensión (que no tiene mucho sentido y hay que tratar de evitarlo salvo sumar un número) toma lo “más natural”.
- (b) Defina un vector x con los números naturales del 1 al 10, y eleve al cuadrado ese vector. ¿Qué hace `R`? ¿ x^2 es igual a $x * x$?
- (c) Aplique alguna operación sobre $q = c(1 : 4227)$ de manera de obtener el vector w del item 7 (d).
10. A los elementos de un vector se les puede poner una etiqueta o nombre de las siguientes maneras:
- (a) `x <- c(a = 1, b = 2, c = 3)`
- (b) `x <- c(1:3) ; names(x) <- c("a", "b", "c")`
- (c) `x <- setNames(1:3, c("a", "b", "c"))`

Vectores de caracteres:

11. Resolver:
- (a) ¿Es posible construir un vector donde la primera coordenada es “mesa”, la segunda es “silla”, y la tercera “mantel” ?, ¿cómo?
- (b) Escriba `letters` en línea de comandos. ¿En qué difiere de `LETTERS`? **Hint:** Estos dos vectores `LETTERS` y `letters` siempre están cargados en `R`.
- (c) ¿Qué longitud tiene el vector `letters`?
- (d) Escriba `paste(letters[1], letters[2], sep = “ ”)` y `paste(letters[1], letters[2], sep = “ ”)` **Hint:** `sep` indica quien separa a las “palabras”. Y suele ser un espacio vacío...

- (e) Escriba en pantalla la palabra “Hola”.
- (f) Escriba en pantalla la oración “Me encanta R” utilizando paste. **Hint:** paste(palabra1, palabra2, palabra3, sep=“ ”)

Números/vectores aleatorios

12. Resolver:


- (a) Escriba `sample(c(1:5),3,replace=T)` y `sample(c(1:5),3,replace=F)`. Vuelva a escribir lo mismo. ¿Qué hace la función sample?
 - (b) Tome 6 números al azar distintos de entre los número del 1 al 100. ¿Guardó los números o los escribió en pantalla?
 - (c) Tome 2 letras al azar del abecedario.
 - (d) Genere una clave numérica aleatoria de 4 letras mayúsculas. **Hint:** paste(vector,collapse=“”)
 - (e) Genere una clave alfanumérica aleatoria de 4 letras mayúsculas y 4 números.
 - (f) ¿Cómo le quedó la clave? ¿Los números y las letras están intercaladas o primero las letras y luego los números? Utilizando los mismos caracteres y números genere una nueva clave permutando las posiciones (coordenadas) del vector. **Hint:** ¿Qué pasa si uno escribe `orden=sample(c(1:10),10,replace=F)`? Comenzamos por `orden = sample(c(1 : *),*,replace = F)`, ¿y ahora?
13. ¿Entendió la diferencia en paste entre `collapse=“”` y `sep=“”`? Corra lo siguiente para sacarse las dudas:

```
> vector <- c(“a”, “b”, “c”)
> paste(vector, collapse=“”)
> paste(vector, sep=“”)
> paste(vector[1],vector[2],vector[3], sep=“”)
```

14. Resolver:


- (a) Tomar una muestra aleatoria de tamaño 1000 de una distribución Normal con media $\mu = 12$ y desvío $\sigma = 2$. **Hint:** `rnorm(*1, *2, *3)`.
- (b) Tomar una muestra aleatoria de tamaño 1000 de una distribución Uniforme entre 3 y 8. **Hint:** `runif(*1, *2, *3)`.
- (c) ¿Qué podría hacer para ver si efectivamente las muestras tomadas tienen la distribución adecuada? **Hint:** Hacer un gráfico, ¿cuál? ¿Le suena el histograma? No se preocupe si no sabe que es un histograma, lo veremos en un par de clases.

Coerción

Muchas veces  “arregla” las cosas a la fuerza sin avisarnos de que hay un error.

15. Defina el siguiente vector, $x < -c(1, 2, 3, 4)$.
 - (a) Verifique que tipo de vector es usando *typeof(x)* o *class(x)*.
 - (b) Cambie el valor de la segunda coordenada por la cadena de caracteres “dos”.
 - (c) Verifique qué tipo de vector es ahora *x*.
16. Defina el siguiente vector, $x < -c(“1”, “2”, “3”, “4”),$ y
 - (a) Verifique qué tipo de vector es.
 - (b) Fuerce a que el vector sea numérico. **Hint:** Mire la tabla de abajo [*as.**]
17. ¿Qué comando convierte un número en un caracter?
18. Coerción producida por nosotros:

Función	Tipo al que hace coerción
<code>as.integer()</code>	Entero
<code>as.numeric()</code>	Numerico
<code>as.character()</code>	Cadena de texto
<code>as.factor()</code>	Factor
<code>as.logical()</code>	Lógico
<code>as.null()</code>	NULL

19. El orden de coerción interna de  es el siguiente:

Las coerciones ocurren en el siguiente orden.



```
lógico -> entero -> numérico -> cadena de texto ( logical -> integer -> numeric -> character )
```

Las coerciones no pueden ocurrir en orden inverso. Podemos coercionar un dato de tipo entero a uno numérico, pero uno de cadena de texto a numérico.

20. Para crear un vector sin una longitud predeterminada escribimos:

```
> x <- vector() # o también x <- c()
# luego podemos ir llenándolo x[1]="a"; x[2]="b"
```

Matrices

21. Cree una matriz de 3x3, con ceros, salvo en la diagonal que contenga unos. *Esta matriz se llama matriz identidad.* **Hint:** Use el comando `matrix(nrow = \star_1 , ncol = \star_2)`.
22. Resuelva el item anterior utilizando los comandos `matrix(0, nrow = \star , ncol = \star)` y `diag(\star_1) = \star_2` .
23. Cree una matriz de 5x2, y complétela (sin hacer loops) con números aleatorios de una muestra de Uniformes[0,1].
- (a) ¿Qué valor está en la coordenada (4,2) de la matriz?
 - (b) ¿Qué dimensión tiene la matriz? **Hint:** `dim(\star)`
 - (c) ¿Es efectivamente una matriz? **Hint:** `class(\star)` o `is.matrix(\star)`
24. ¿Entiende la diferencia entre `is.matrix(\star)` y `as.matrix(\star)`?
25. Cree dos matrices de 4x3, y complete cada una con números aleatorios de una muestra de Uniformes[0,1].
- (a) ¿Qué da la suma las dos matrices?
 - (b) ¿Y el producto? *Esto lo hace  pero el producto entre matrices no es este.*
 - (c) ¿Y la división? *Esto lo hace  pero no existe la división de matrices.*
 - (d) ¿Es efectivamente una matriz?
26. Cree dos matrices de 3x2, y complete cada una de ellas con números aleatorios de una muestra de Uniformes[0,1].
- (a) Construya la matriz transpuesta, es decir la matriz de 2x3 con los mismos elementos de la anterior. **Hint:** `t(\star)`
 - (b) Haga el producto entre las matrices anteriores. ¿Se puede hacer el producto anterior (coordenada a coordenada)?
 - (c) Haga el producto matricial entre las matrices anteriores. **Hint:** cambiar `*` por `% * %`. **LO DISCUTIMOS JUNTOS**
 - (d) ¿Qué dimensión tiene la matriz del producto matricial?
 - (e) ¿Da lo mismo hacer el producto matricial de una contra la otra que viceversa? ¿Qué dimensión tiene cada una de las matrices producto?

27. Cree una matriz de 4×2 , y complete cada una de ellas con la secuencia del 1 al 8. ¿Qué criterio tiene `R` para completar la matriz con `c(1:8)`? ¿Completa filas o columnas primero?
- (a) Haga el producto matricial de la matriz creada con su traspuesta. ¿Qué valor espera obtener en el lugar `[3,2]`? ¿Lo hace bien `R`?
- (b) Entonces, ¿una matriz de 4×2 multiplicada por una 2×6 qué dimensión tiene? ¿y una de $n_1 \times m$ multiplicada por una de $m \times n_2$?
28. ¿Se puede multiplicar una matriz de 4×3 por una matriz de 3×1 ? ¿Qué es una matriz de 3×1 ? ¿Qué dimensión tiene ese producto?
29. **Norma de un vector.** Sea $x \leftarrow c(1,2)$, ¿cuál es la longitud del vector x ? ¿Es decir, ¿cuál es la longitud de la flecha que sale del punto $(0,0)$ y termina en el punto $(1,2)$?
30. Sea $x \leftarrow c(1:1000)$, ¿cuál es la norma o longitud del vector x ?
31. ¿Cuántos vectores diferentes de dimensión dos se pueden armar con las letras “a”, “b” y “c”? Ejemplo: `c(“a”, “c”)` otro `c(“c”, “a”)` otro `c(“b”, “a”)`. Para responder esta pregunta utilice el comando `permutations(,)` de la librería `gtools`. Primero construya un vector con las letras “a”, “b” y “c” y luego use la función `permutations`
- (a) A partir del comando `class()` confirme que el comando `permutations` devuelve una matriz.
- (b) Observar la diferencia entre: `x=c(“a”, “b”, “c”);`
`permutations(n=3,r=2,v=x,repeats.allowed=F);`
`permutations(n=3,r=2,v=x,repeats.allowed=T)`

Arrays

32. Corra lo siguiente: `x<-array(dim=c(2,3,5))`

- (a) ¿Qué significa cada una de las dimensiones? ¿x tiene 5 matrices de cuánto por cuánto? ¿Cuántos números “casilleros” se pueden llenar?
- (b) Complete a x con números del 1 al 30. ¿Cómo los completa por defecto?
- (c) ¿Qué número está en la posición 2,3,5? Cambie ese valor por el número 0

Listas

33. Las listas permiten cargar de todo un poco... Es la estructura más flexible. Las listas se usa doble corchete `[[]]`. Escribir:

```
> x<- list()
> x[[1]]<-c(“a”, “b”)
> x[[2]]<-c(1:10)
> x
```

34. Otro ejemplo:

```
> ejemplolista <- list(nombre=“Pedro”, casado=T,
  esposa=“María”,no.hijos=3, edad.hijos=list(c(4,7,9)))
```

35. Otro ejemplo:

```
> x=list(nombre=c(“Pedro”, “Maria”), casado=c(T,F),
  conyuge=c(“Romina”, “Juan”),no.hijos=c(4,1), edad.hijos=list(list(c(4,7,9),21),list(3)))
```

- (a) Explore las diferencia entre `x$edad.hijos[[1]][1]` y `x$edad.hijos[[1]][1]`. **Hint:** `is.list()`.
- (b) Calcule la edad promedio de los hijos que tuvo Pedro con Romina.

36. Otro ejemplo:

```
> vec <- 1:10
> mat <- matrix(1:4, nrow = 2,ncol=2)
> df <- data.frame(“num” = 1:3, “letras” = c(“a”, “b”, “c”))
> lista <- list(“x” = vec, “y” = mat, “z” = df)
```

37. Comentario. El comando `unlist()` convierte una lista en un vector. Pero ojo porque pocas veces puede ser una buena estrategia pasarlo a vector. Además recordar que los vectores son atómicos y las listas no.

Data Frame

38. Son planillas. En una misma estructura podemos tener columnas con números y otras con caracteres. Es decir, no son objetos atómicos (`is.atomic()`). Construya los data frames de abajo y observe las tablas.

- (a) `datos <- data.frame(edad=c(22,21,34,42,17,43),
tipo=c("+","+","-","+","-","+"),sexo=c("H","M","H","H","M","H"))`
- (b) `df1 <- data.frame(x = 1:3, y = c("a", "b", "c"), stringsAsFactors = FALSE)`
- (c) `df2 <- data.frame(x = 1:3, y = c("a", "b", "c"), stringsAsFactors = TRUE)`
- (d) Observar la diferencia entre `df1$y` y `df2$y`.
- (e) `df3 <- data.frame(age = c(35, 27, 18), hair = c("blond", "brown", "black"),
row.names = c("Bob", "Susan", "Sam"))`
- (f) `df4<-data.frame(x = 1:3, y = I(list(1:2, 1:3, 1:4)))`
- (g) `df5<-data.frame(nombre=c("Juan","Maria","Pedro"),edad=c(22,21,34),corona=c("-
","-","+"), sexo=c("H","M","H"),
hijos=c(0,2,1),edad.hijos=I(list(NULL,c(1,3),3)))`

39. Calcule la edad promedio de los hijos de María de la tabla `df5`.

Factores

Variables categóricas: muchas veces tenemos variables que toman valores que son categorías. Supongamos que estudiamos la variable *grupo sanguíneo* en una muestra y construimos un vector con esta información donde cada coordenada representa el grupo sanguíneo de cada persona. Hay dos maneras de construir este vector:

- (a) `grup_sang_char <-c("0","0","B", "0", ..., "A")`
- (b) `grup_sang_factor <- factor(grup_sang_char, levels=c("A","B", "AB", "0"))`

La diferencia entre `grup_sang_char <` y `grup_sang_factor` radica en que `grup_sang_factor` va a tener todos los valores que puede tomar la variable sin importar si fueron observados en la muestra. Por lo tanto cuando estudiemos la variable factor algunas categorías van a tener cero observaciones. En cambio para la variable carácter directamente no van a existir. También podemos poner en la variable factor solamente las categorías que aparecen y esto ayudará al momento de analizar y graficar ya que muchas veces es necesario tener una variable factor.

40. Coerción para tipo de dato:

Función	Coerciona a	Coerciona exitosamente a
<code>as.vector()</code>	Vector	Matrices
<code>as.matrix()</code>	Matrices	Vectores, Data frames
<code>as.data.frame()</code>	Data frame	Vectores, Matrices
<code>as.list()</code>	Lista	Vectores, Matrices, Data frames

EJERCICIO INTEGRADOR PARA ENTREGAR

41. Lea el ejercicio completo antes de resolver.

- Investigue qué hacen las funciones `grep()`, `strsplit()`
- Con todos las categorías de identidad de género que se encuentran en el post *Sexualities master list* de la página web “<https://www.asexuality.org/en/topic/123256-asexuality-sexual-orientation-lexicon-read-me/>” construya un dataframe donde una de las variables sea el factor identidad de género (I_G) y la otra sea la descripción de la categoría (DESCRIP).
- Para hacer el punto (b) comience con los siguientes comandos:


```
> url="https://www.asexuality.org/en/topic/123256-asexuality-sexual-orientation-lexicon-read-me/"
> pagina = readLines(url) # ya tenemos toda la info de la pagina
> # ahora toca arremangarse las manos. Hint: ¿Qué objeto es pagina?
# puede usar las funciones grep(), strsplit(), paste(), unlist()
# NO puede usar for , library(stringr)
# Mire la página desde algún browser para entender la info y no perderse
# Piense en forma vectorial
```
- Tome una muestra aleatoria con repetición de 30 elementos de la variable I_G anterior donde cada una de las categorías tiene la misma chance. Supongamos que estos valores representan los valores observados en una muestra. Llame a este vector *muestra*.
- Utilizando el comando `table` y el comando `barplot` estudie la distribución de *muestra*.
- Convierta a el vector *muestra* en un factor que incluya todas las categorías existentes en el dataframe. Repita el punto (e) sobre ahora la muestra como un factor. ¿Hay alguna diferencia?