

Introduccion R

Estefania Mancini

Marzo 2022

Introduccion a R

- Directorio de trabajo: Informar sobre el directorio de trabajo actual
- Cambiar el directorio de trabajo a la carpeta **dia2** (hay que crearla)
- Informar la versión de R con la que está trabajando y los paquetes instalados:
- Usar la ayuda (y por qué no google :)) para hallar información sobre:

** la varianza de una variable random

** La raiz cuadrada de un numero:

Informacion acerca del paquete “graphics”

Las funciones mean y median

Analice con cuidado los parametros que son obligatorios y la resolucion cuando hay datos faltantes (NA)

- Obtenga una lista de todas las funciones en R que contienen la cadena de caracteres test en su nombre.

Guardar objetos

- Guarde todos los comandos ejecutados y sus resultados de esta sesión en el archivo “Primera.R”. Si no sabe como hacerlo revise la ayuda de save, history.save

(en el menú principal, seleccionar Archivo -> Guardar en Archivo)

Si quiero guardar los objetos, hay que hacer:

y luego

- Salga de RStudio sin guardar el área de trabajo (workspace) y vuelva a ingresar y abra el script guardado.

Generando datos

- Cree el vector **info** que contiene su edad, altura (en cm) y código postal.
- Obtener el cuadrado de un numero:
- Obtener el valor de la constante pi. Cuanto vale pi? Como lo sabe R?
- Obtener la tangente de un angulo:
- Recalcule usando el coseno. ¿dónde se guardan estos resultados?
- Asignar el valor de pi a una variable a1, luego el coseno de pi.

*Asignar a B el valor de -3: y crear una nueva variable que reduzca su valor en -2.5.

Probar la asignacion usando espacio entre el < y el -:

- Asignar el valor de -1000 a pi y volver a obtener su valor:
- Tipear `base::pi` y compara los resultados con lo anterior.

VECTORES

- Definir un vector numerico n como:
- Obtener el minimo (min), maximo (max), suma (sum), media (mean) y varianza (var) de n.
- Definir un vector logico p y un vector caracter a. Aplicar la funcion `class()` a n, p y q
- La función secuencia `seq()`: permite crear una cadena de valores con cualquier incremento que se desee. Se puede especificar el valor incremental o la longitud deseada:
- Analice con la ayuda la funcion `seq` y cree un vector, de longitud 10, con separaciones cada 0.1

(No subestime esta funcion, es muy util para generar intervalos para histogramas y labels para plots.)

- La funcion replicar `rep()`: permite repetir un valor o una secuencia de valores una cantidad especificada de veces:
- Subsetear

Definir un vector numerico I como: `i <- -1:-10`

Usando un vector de indexacion de los posibles enteros, definir unvector i2 que contiene el 2do, 5to y 7mo elemento del vector i usando un vector de indexacion de enteros negativos, definir un vector i3 que contiene el vector 1 sin el 1er, 3er y 9no elemento

Usar un vector condicional i4 como indice que contiene solo los elementos de I que son mayores a -5

- Realizar las siguientes operaciones

Que pasa con z que tiene diferente longitud?

- Uso de valores NA y NaN: Generar una variable como `c(1:3)`.

Que se obtiene si intento extraer la 4ta posicion de esta variable? Como determino si una variable tiene valores ausentes?

- Misc vectores:

Crear un vector numerico y asignarlo a x. Luego ordenar de manera creciente y decreciente y guardarlos en cx, dx. `sort()`, `order()`

Crear un vector numérico h aplicar la funcion `prod(h)`

Crear un vector numérico z y aplicar la funcion `summary(z)`

Concatenar z como caracter y aplicar la funcion `summary()`.

Crear un vector logico L y coercionarlo a valores numéricos usando `as.numeric()` y `as.double()`

Usar la funcion `paste()` para concatenar z y “es un caracter”. No subestimar la funcion `paste`, es muy util por ejemplo para armar nombres de archivos o de variables de forma dinámica

Matrices

Entre las poderosas características de R se encuentra su habilidad para realizar operaciones matriciales. Se pueden crear objetos matriciales a partir de un vector numérico utilizando el comando `matrix()`:

- Otra forma de generar matrices a partir de vectores es utilizando las funciones `cbind()` y `rbind()`

- Repetir el cbind con:

La función `rbind()` pega sus argumentos por filas. Obtenemos también en este caso un objeto de clase `matrix`.

- Definir una matriz M1 de 10x10, con los datos de la secuencia de 101 a 200
- Definir una submatriz M2 que contiene todas las filas pero solo las primeras 5 columnas de M1
- Definir una submatriz M3 igual a M1 exceptuando la toda la fila y columna 5.
- Aplicar la función `dim()` a todas las matrices (M1, M2, M3)

Listas

Definir una matriz numérica, un vector de carácter y un vector lógico. Construir una lista L1 con los 3 componentes dando a cada uno un nombre de elección.

Definir un vector factorial y un vector numérico. Construir una lista L2 con estos 2 componentes y asignarles un nombre.

Combinar las listas L1 y L2 en una tercera L3.

Aplicar las funciones `class()` y `attributes()` a L3 como un todo y a cada uno de sus componentes.

Objetos

- Listar todos los objetos y variables que están cargados en su sesión actual.
- Remover algunos objetos/variables a su elección

Paquetes

- Listar todos los paquetes instalados y cargados
- Chequear si están instalados los paquetes: “knitr” y “markdown”. Si es así, cargarlos. Si no, instalarlos.
- Quitar los paquetes del entorno
- Cargar un paquete de su preferencia y chequear una función que este posea antes y después de cargarlo. (ejemplo: paquete `limma`, función `VennDiagram`)

*Instalar el paquete “xpose4” usando el comando `install.packages()`.

- Datos de paquetes instalados: R contiene muchos conjuntos de datos incorporados al software. Estos datos están guardados como data frames. Para ver la lista de conjuntos de datos tipee `data()`. Se abrirá una ventana en la que se listan los conjuntos de datos disponibles (hay muchos otros en los paquetes escritos por usuarios). Para saber algo más sobre algún conjunto de datos disponible tipee:
- Cargar los datos del paquete `iris` y conocer su clase. También determinar la clase de los componentes “Sepal.Length” y “Species”.

Clases

Convertir de un tipo de dato a otro

- Para acceder a las variables individuales en un data frame, se usa un `$` entre el nombre del data frame y el nombre de la columna:

También se puede acceder a un elemento individual o a una fila especificando su posición (en un formato fila columna) entre corchetes a continuación del nombre del data frame:

Input/Output

Cree un data frame llamado cronograma con las variables: nombre,días, cantidad de horas semanales. Genere un archivo de datos utilizando una planilla de cálculo que contenga las variables: edad, género, cantidad de materias aprobadas de un curso hipotético con 15 alumnos. Incluya el nombre de las variables en la primera fila. Guárdelo en formato texto (por ejemplo, delimitado por tabulaciones) e impórtelo a un data frame del R. Guarde en un data frame llamado tempacid las variables Water.Temp y Acid.Conc. del dataset stackloss de R.

Bonus:

Podríamos armarlo con algun google docs sobre las carreras, edades, origen, etc... de las asistentes, o sobre algun dato que nos interese.

Uso de comentarios

Ingresar un comentario en la linea de comando:

Funciones

Crear una funcion:

f1: que devuelva la suma de las medias de 2 vectores. Guardarla en un archivo f1.R y cargarlo en una nueva sesion de R usando `source("")`

f2: acepte 3 vectores numericos y devuelva el producto de su longitud dividido por el valor del mas largo. Guardarla en un archivo f2.R y cargarlo en una nueva sesion de R usando `source("")`

f3: Crear una funcion que acepte un vector y una variable y devuelva el valor de $\text{vector}^{\text{variable}}$. Si no se provee la variable, se devuelve el valor del vector^2