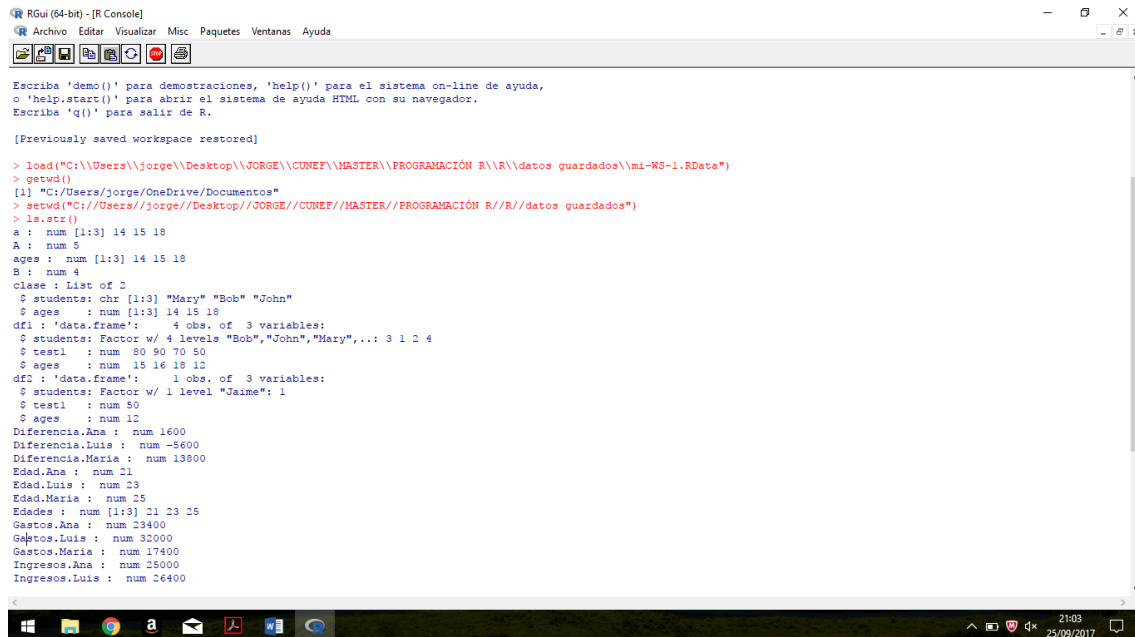


Ejercicio 1. RECUPERAR WORKSPACE

Ejecutar `getwd()` para conocer dónde está el workspace del sistema y comprobado que se recupera el almacenado en la práctica 1. Utilizar `setwd("ruta de la carpeta")` para asignar la carpeta donde se desea que se guarde el workspace que se va a crear en la sesión o `load("ruta de la carpeta")` para cargar la que se desee en cada momento. Es recomendable una carpeta por cada práctica o trabajo, para tener diferenciados las tareas realizadas. Se puede utilizar el sistema de menús de R y de RStudio para este cometido.



```

RGui (64-bit) - [R Console]
Archivo  Editor  Visualizar  Misc  Paquetes  Ventanas  Ayuda

Escriba 'demo()' para demostraciones, 'help()' para el sistema on-line de ayuda,
o 'help.start()' para abrir el sistema de ayuda HTML con su navegador.
Escriba 'q()' para salir de R.

[Previously saved workspace restored]

> load("C:\\Users\\jorge\\Desktop\\JORGE\\CUNEF\\MASTER\\PROGRAMACIÓN R\\R\\datos guardados\\mi-WS-1.RData")
> getwd()
[1] "C:/Users/jorge/OneDrive/Documentos"
> setwd("C://Users//jorge//Desktop//JORGE//CUNEF//MASTER//PROGRAMACIÓN R//R//datos guardados")
> ls.str()
a : num [1:3] 14 15 18
A : num 5
ages : num [1:3] 14 15 18
B : num 4
class : List of 2
 $ students: chr [1:3] "Mary" "Bob" "John"
 $ ages : num [1:3] 14 15 18
df1 : 'data.frame': 4 obs. of 3 variables:
 $ students: Factor w/ 4 levels "Bob","John","Mary",...: 3 1 2 4
 $ test1 : num 80 90 70 50
 $ ages : num 15 16 18 12
df2 : 'data.frame': 1 obs. of 3 variables:
 $ students: Factor w/ 1 level "Jaime": 1
 $ test1 : num 50
 $ ages : num 12
Diferencia.Ana : num 1600
Diferencia.Luis : num -5600
Diferencia.Maria : num 13800
Edad.Ana : num 21
Edad.Luis : num 23
Edad.Maria : num 25
Edades : num [1:3] 21 23 25
Gastos.Ana : num 23400
Gastos.Luis : num 32000
Gastos.Maria : num 17400
Ingresos.Ana : num 25000
Ingresos.Luis : num 26400
  
```

SCRIPT

```

### EJERCICIO 1 ###

load("C:\\Users\\jorge\\Desktop\\JORGE\\CUNEF\\MASTER\\PROGRAMACIÓN R\\R\\datos guardados\\mi-WS-1.RData")
getwd()
setwd("C://Users//jorge//Desktop//JORGE//CUNEF//MASTER//PROGRAMACIÓN R//R//datos guardados")
ls.str()
  
```

Ejercicio 2. ESTRUCTURAS DE DATOS

Creación de Data Frames. Un data.frame es una estructura donde almacenar objetos de diferentes tipos. En este ejercicio queremos crear una estructura por cada una de las personas de la tabla del ejercicio 2 de la práctica 1. Tras realizar la asociación, comprobar el resultado visualizando el contenido de workspace.

Para ello, creamos un data.frame con la sentencia data.frame(lista de objetos separada por comas). Análogamente lo podremos hacer con data.table o transformar directamente dt<-data.table(objeto tipo data.frame), de esta forma el nuevo objeto dt tendrá habilitadas todas las funciones de data.table, comprobad con str(dt)

```
R Console
MediaEdades : num 23
MediaGastos : num 24267
MediaIngresos : num 27533
n.temps : num [1:4] 13 15 15 17
Nombre.Ana : chr "Ana"
Nombre.Luis : chr "Luis"
Nombre.Maria : chr "Maria"
Numeros : num [1:12] 1 2 3 4 5 6 7 8 9 10 ...
Primos : num [1:9] 2 3 5 7 11 13 17 19 23
s : chr [1:3] "Mary" "Bob" "John"
s.edades : logi [1:6] TRUE FALSE TRUE TRUE FALSE TRUE
scores : num [1:2, 1:3] 1 2 80 85 67 56
students : chr [1:3] "Maria" "Bob" "John"
temps : num [1:4] 10 11 12 13
x : num [1:12] 0 1 2 3 4 5 6 7 8 9 ...
xm : num 8.75
> save.image("C:\\Users\\jorge\\Desktop\\JORGE\\CUNEF\\MASTER\\PROGRAMACIÓN R\\R\\datos guardados\\mi-WS-PRACTICA2.RData")
> dt.Ana <- data.frame(Edad.Ana, Gastos.Ana, Ingresos.Ana); dt1 <- (dt.Ana)
> dt.Luis <- data.frame(Edad.Luis, Gastos.Luis, Ingresos.Luis); dt2 <- (dt.Luis)
> dt.Maria <- data.frame(Edad.Maria, Gastos.Maria, Ingresos.Maria); dt3 <- (dt.Maria)
> dt.Ana <- data.frame(Edad.Ana, Gastos.Ana, Ingresos.Ana); dt1 <- (dt.Ana); str(dt.Ana)
'data.frame': 1 obs. of 3 variables:
 $ Edad.Ana : num 21
 $ Gastos.Ana : num 23400
 $ Ingresos.Ana: num 25000
> dt.Luis <- data.frame(Edad.Luis, Gastos.Luis, Ingresos.Luis); dt2 <- (dt.Luis); str(dt.Luis)
'data.frame': 1 obs. of 3 variables:
 $ Edad.Luis : num 23
 $ Gastos.Luis : num 32000
 $ Ingresos.Luis: num 26400
> dt.Maria <- data.frame(Edad.Maria, Gastos.Maria, Ingresos.Maria); dt3 <- (dt.Maria); str(dt.Maria)
'data.frame': 1 obs. of 3 variables:
 $ Edad.Maria : num 25
 $ Gastos.Maria : num 17400
 $ Ingresos.Maria: num 31200
> |
```

SCRIPT

```
### EJERCICIO 2 ###
```

```
dt.Ana <- data.frame(Edad.Ana, Gastos.Ana, Ingresos.Ana); dt1 <- (dt.Ana); str(dt.Ana)
dt.Luis <- data.frame(Edad.Luis, Gastos.Luis, Ingresos.Luis); dt2 <- (dt.Luis); str(dt.Luis)
dt.Maria <- data.frame(Edad.Maria, Gastos.Maria, Ingresos.Maria); dt3 <- (dt.Maria); str(dt.Maria)
```

Ejercicio 3. CREACIÓN DE SCRIPTS

Crear un script mediante el sistema de menús de R llamado “minombreP2.R”.

Dentro del script, escribid los comentarios tras el símbolo # que anula el resto de la línea del proceso de ejecución. Añadid un título con vuestro nombre completo.

En cada línea del script debéis escribir una sentencia que posteriormente podrá ser ejecutada cuantas veces se desee. Se pide escribir líneas para ejecutar las siguientes sentencias:

- Calcular $2 \times A - B$ sin almacenar en ningún otro objeto
- Calcular lo mismo almacenando el resultado en un nuevo objeto denominado C
- Ejecutar `typeof(C)` para conocer el tipo que R ha asignado automáticamente al nuevo objeto.
- Calcular los números pares de la lista `Numeros` mediante discriminación por rango y almacenar el resultado en el objeto `Pares`. Hacer lo mismo para construir el objeto `Impares`.
- Comprobar con `ls()` el contenido de workspace y con `ls.str()` los tipos y los contenidos de los objetos creados.
- Salvar el workspace y el script. Cerrar R.



```
> #JORGE FUERTES
>
> 2*A-B
[1] 6
> C <- 2*A-B
> typeof(C)
[1] "double"
> Numeros
[1] 1 2 3 4 5 6 7 8 9 10 11 12
> Pares <- Numeros[seq(from=2, to=12, by=2)]; Pares
[1] 2 4 6 8 10 12
> Impares <- Numeros[seq(from=1, to=11, by=2)]; Impares
[1] 1 3 5 7 9 11
> ls()
[1] "a"          "A"          "ages"       "B"          "C"          "clase"
[7] "df1"       "df2"       "Diferencia.Ana" "Diferencia.Luis" "Diferencia.Maria" "dt.Ana"
[13] "dt.Luis"   "dt.Maria"  "dt1"        "dt2"        "dt3"        "Edad.Ana"
[19] "Edad.Luis" "Edad.Maria" "Edades"     "Gastos.Ana"  "Gastos.Luis"  "Gastos.Maria"
[25] "Impares"   "Impares1"   "Ingresos.Ana" "Ingresos.Luis" "Ingresos.Maria" "MediaEdades"
[31] "MediaGastos" "MediaIngresos" "n.temps"    "Nombre.Ana"  "Nombre.Luis"  "Nombre.Maria"
[37] "Numeros"   "Pares"      "Pares1"     "Primos"      "s"            "s.edades"
[43] "scores"    "students"   "temps"     "x"           "xm"
> ls.str()
a : num [1:3] 14 15 18
A : num 5
ages : num [1:3] 14 15 18
B : num 4
C : num 6
clase : List of 2
 $ students: chr [1:3] "Mary" "Bob" "John"
 $ ages : num [1:3] 14 15 18
df1 : 'data.frame': 4 obs. of 3 variables:
 $ students: Factor w/ 4 levels "Bob","John","Mary",...: 3 1 2 4
 $ test1 : num 80 90 70 50
 $ ages : num 15 16 18 12
df2 : 'data.frame': 1 obs. of 3 variables:
```

SCRIPT

```
### EJERCICIO 3 ###

#JORGE FUERTES

2*A-B
C <- 2*A-B
typeof(C)
Numeros
Pares <- Numeros[seq(from=2, to=12, by=2)]; Pares
Impares <- Numeros[seq(from=1, to=11, by=2)]; Impares
ls()
ls.str()
```

Ejercicio 4.

Abrir R y recuperar el trabajo del ejercicio anterior. Comprobad el contenido de las variables.

Teniendo en cuenta que la variable `s.edades` representa el sexo (femenino =True) de la persona a la que corresponde la edad del vector `Edades`, se desea crear un objeto `F.edades` donde almacenar las edades de las personas de sexo femenino. Añadir el comando al script de trabajo. Utilizar el comando `print()` para comprobar el resultado. Utilizar el comando `seq` para crear una secuencia de números del 3 al 7 y almacenarla en el objeto "secuencia".

Ejecutar cada una de las sentencias siguientes y comentar el resultado:

```
>Primos[secuencia]; Primos[secuencia-1]; Primos[secuencia-2]; secuencia.siguiente<-
secuencia+6; secuencia.siguiente; Primos[secuencia.siguiente]
```

```
[19] "Edad.Luis"      "Edad.Maria"      "Edades"
[22] "Gastos.Ana"    "Gastos.Luis"     "Gastos.Maria"
[25] "Impares"       "Impares1"        "Ingresos.Ana"
[28] "Ingresos.Luis" "Ingresos.Maria"  "MediaEdades"
[31] "MediaGastos"   "MediaIngresos"   "n.temps"
[34] "Nombre.Ana"    "Nombre.Luis"     "Nombre.Maria"
[37] "Numeros"       "Pares"           "Pares1"
[40] "Primos"        "s"               "s.edades"
[43] "scores"        "secuencia"       "secuencia.siguiente"
[46] "students"      "temps"           "x"
[49] "xm"

> F.Edades <- Edades(s.edades); print(F.Edades)
Error in Edades(s.edades) : no se pudo encontrar la función "Edades"
> secuencia <- Numeros[seq(from=3, to=7, by=1)]; secuencia
[1] 3 4 5 6 7
> # Creamos un vector con el mismo n° de componentes que el de la secuencia, empezando por el siguiente
> Primos[secuencia]
[1] 5 7 11 13 17
>
> # Creamos un vector con el mismo n° de componentes que el de la secuencia, empezando por el 3(inicio
> Primos[secuencia-1]
[1] 3 5 7 11 13
>
> # Creamos un vector con el mismo n° de componentes que el de la secuencia, empezando por el n° ant
> Primos[secuencia-2]
[1] 2 3 5 7 11
>
> # Creamos secuencia sumando 6 a cada elemento de la misma:
> secuencia.siguiente<-secuencia+6; secuencia.siguiente
[1] 9 10 11 12 13
>
> # Creamos un vector con el mismo n° de elementos que el vector secuencia.
> # Busca los numeros primos sumando 6 al vector primos, por lo que el primer componente es 23 y el
> # ya que el vector primos no tiene mas componentes definidos.
> Primos[secuencia.siguiente]
[1] 23 NA NA NA NA
```

SCRIPT

```
### EJERCICIO 4 ###

ls()
F.Edades <- Edades(s.edades); print(F.Edades)
secuencia <- Numeros[seq(from=3, to=7, by=1)]; secuencia
# Creamos un vector con el mismo n° de componentes que el de la secuencia, empezando por el siguiente primo de 3(inicio de la secuencia):
Primos[secuencia]

# Creamos un vector con el mismo n° de componentes que el de la secuencia, empezando por el 3(inicio de la secuencia):
Primos[secuencia-1]

# Creamos un vector con el mismo n° de componentes que el de la secuencia, empezando por el n° anterior a la secuencia:
Primos[secuencia-2]

# Creamos secuencia sumando 6 a cada elemento de la misma:
secuencia.siguiente<-secuencia+6; secuencia.siguiente

# Creamos un vector con el mismo n° de elementos que el vector secuencia.
# Busca los numeros primos sumando 6 al vector primos, por lo que el primer componente es 23 y el resto no están definidas
# ya que el vector primos no tiene mas componentes definidos.
Primos[secuencia.siguiente]
```