



Módulo 4: Parte I

Solución de los Ejercicios

Autor: Raquel Dormido Canto

Actualizado Enero 2023

Contenido

EJERCICIO 1.....	3
EJERCICIO 2.....	5
EJERCICIO 3.....	5
EJERCICIO 4.....	7
EJERCICIO 5.....	9
EJERCICIO 6.....	11
EJERCICIO 7.....	13
EJERCICIO 8.....	13
EJERCICIO 9.....	14

EJERCICIO 1.

Copia o teclea las líneas de código que aparecen a continuación. Cada una de las líneas en el prompt, una por una, y ejecútalas, pulsando Entrar. Intenta adivinar el resultado de cada operación antes de ejecutar.

```
3+4
13-6
5*4
13/5
13%/%5
13%%5
1/3+1/5
sqrt(9)
sin(pi)
sin(3.14)
```

Guarda un pantallazo de las operaciones realizadas con sus resultados. A la vista de los resultados ¿trabaja R con representación mediante decimales? ¿Puede haber pérdida de precisión en determinadas operaciones de R? Comenta como interpreta R la operación $1/3+1/5$. ¿La interpreta como $\frac{1}{3} + \frac{1}{5}$ o como $\frac{1}{(\frac{3+1}{5})}$? ¿El valor obtenido para $\sin(\pi)$ es el esperado? ¿Qué diferencia encuentra entre $\sin(\pi)$ y $\sin(3.14)$?

Comentarios al Ejercicio 1: Los operadores $\%/\%$ y $\%\%$ representan, respectivamente, el cociente y el resto de la división entera. La interpretación que realiza R de una expresión es debida a la prioridad de operadores que tiene definida el lenguaje. La función `sin` es la función seno de trigonometría. Por π R denota la constante matemática $\pi \sim 3.141593$.

Solución Ejercicio 1

```
> 3+4
[1] 7
> 13-6
[1] 7
> 5*4
[1] 20
> 13/5
[1] 2.6
> 13%/%5
[1] 2
> 13%%5
[1] 3
> 1/3+1/5
[1] 0.5333333
```

```
> sqrt(9)
[1] 3
> sin(pi)
[1] 1.224606e-16
> sin(3.14)
[1] 0.001592653
```

Lo más importante que tienes que observar en los resultados es que R trabaja siempre con la representación mediante decimales de los números. Esto es evidente, por ejemplo, en la respuesta a la instrucción:

```
1/3+1/5
```

que representa la operación

$$\frac{1}{3} + \frac{1}{5}$$

pero cuyo resultado en R es 0.5333333 y no 8/15. El valor decimal es una aproximación a la fracción, pero no es un resultado exacto (la fracción sí es exacta). Por lo tanto, cuando hagamos operaciones con R, tenemos que tener cuidado con la posible pérdida de precisión, que llevan siempre aparejadas las operaciones con decimales.

Fíjate en que, como hemos dicho, R ha interpretado el símbolo $1/3+1/5$.

como la operación $\frac{1}{3} + \frac{1}{5}$ en lugar de darle otras interpretaciones posibles como, por ejemplo $\frac{1}{\left(\frac{3+1}{5}\right)}$.

Para hacer esa interpretación R ha aplicado una serie de reglas, de lo que se conoce como prioridad de operadores, y que dicen en qué orden se realizan las operaciones, según el tipo de operador. Siempre se pueden utilizar los paréntesis para evitar posibles ambigüedades en las expresiones. Podríamos escribir en este caso: $(1/3) + (1/5)$ o bien $1 / ((3+1) / 5)$

Y el carácter numérico de R se refleja de nuevo en el hecho de que al ejecutar `sin(pi)` no se obtiene 0 (que es la respuesta exacta), sino:

```
> sin(pi)
[1] 1.224606e-16
```

La notación que se usa en la respuesta es la forma típica de traducir la notación científica a los lenguajes de ordenador y calculadoras. El símbolo $1.224606e-16$ representa el número:

1.224606×10^{-16}

de manera que el número -16 , que sigue a la letra e en esta representación, es el exponente de 10 (también llamado orden de magnitud), mientras que el número 1.224606 se denomina mantisa. En cualquier caso, el exponente -16 nos indica que se trata de un número extremadamente cercano a 0. Fíjate, en cambio, que si usas 3.14 como aproximación de π , la respuesta, aunque pequeña, es todavía del orden de milésimas.

EJERCICIO 2.

¿Qué se obtiene al ejecutar estos comandos, uno detrás de otro? ¿Cuánto valen las variables a , b y c al final?

```
a = 2
b = 3
c = a + b
a = b * c
b = (c - a)^2
c = a * b
```

Solución Ejercicio 2

Al final las variables valen $a = 15$, $b = 100$, $c = 1500$.

EJERCICIO 3.

Supongamos que tenemos un vector con la población de origen de 15 estudiantes. Este vector debe contener como información la siguiente:

getafe, mostoles, madrid, madrid, mostoles, leganes, getafe, leganes, madrid, mostoles, parla, alcorcon, mostoles, getafe, leganes

Además disponemos de las estaturas de cada uno de los estudiantes, que es la siguiente:

1.83, 1.71, 1.79, 1.64, 1.74, 1.81, 1.62, 1.84, 1.68, 1.81, 1.82, 1.74, 1.84, 1.61, 1.84

- Mostrar los niveles del factor (las poblaciones de origen), junto con el número de estudiantes correspondientes a tales niveles.
- Calcular la estatura promedio de los estudiantes de cada población a partir de la muestra de la que disponemos.

Nota: En el apartado a) hacer uso de la función `factor` y de la función `summary`. En el apartado b) hacer uso de la función `tapply` y de la función `mean`

Solución Ejercicio 3

En primer lugar se crea el vector que contiene las 15 poblaciones de origen de los estudiantes. En la segunda orden se muestran dichos orígenes. En la tercera se pregunta la longitud de dicho vector:

```
> estudiantes.origen = c("getafe","mostoles","madrid","madrid","mostol
es","leganes","getafe","leganes","madrid","mostoles","parla","alcorcon
","mostoles","getafe","leganes")
> estudiantes.origen
[1] "getafe" "mostoles" "madrid" "madrid" "mostoles" "leganes"
[7] "getafe" "leganes" "madrid" "mostoles" "parla" "alcorcon"
[13] "mostoles" "getafe" "leganes"
> length(estudiantes.origen)
[1] 15
```

Ahora creamos una variable de tipo factor, a partir de la existente:

```
> festudiantes = as.factor(estudiantes.origen)
> festudiantes
[1] getafe mostoles madrid madrid mostoles leganes getafe leganes
[9] madrid mostoles parla alcorcon mostoles getafe leganes
Levels: alcorcon getafe leganes madrid mostoles parla
> levels(festudiantes)
[1] "alcorcon" "getafe" "leganes" "madrid" "mostoles" "parla"
> summary(festudiantes)
alcorcon getafe leganes madrid mostoles parla
      1         3         3         3         4         1
```

Al pedir un sumario de la variable de tipo factor 'festudiantes', el resultado es una tabla que nos muestra los niveles del factor (las poblaciones de origen), junto con el número de estudiantes correspondiente a tales niveles.

Supongamos ahora que disponemos de las estaturas de cada uno de los estudiantes del ejemplo anterior:

```
> estudiantes.estaturas = c(1.83, 1.71, 1.79, 1.64, 1.74, 1.81, 1.62,
1.84, 1.68, 1.81, 1.82, 1.74, 1.84, 1.61, 1.84)
> estudiantes.estaturas
```

```
[1] 1.83 1.71 1.79 1.64 1.74 1.81 1.62 1.84 1.68 1.81 1.82 1.74 1.84
1.61 1.84
```

Vamos a calcular ahora la estatura promedio de los estudiantes de cada población a partir de la muestra de la que disponemos:

```
>tapply(estudiantes.estaturas,festudiantes,mean)
alcorcon getafe leganes madrid mostoles parla
1.740000 1.686667 1.830000 1.703333 1.775000 1.820000
```

La función **tapply()** se utiliza para aplicar una función, en este caso **mean()** para cada grupo de componentes del primer argumento, definidos por los niveles de la segunda componente, en este caso, **festudiantes**.

EJERCICIO 4.

Construir una matriz de 14x3 donde los nombres de las columnas son las variables peso, altura y edad de 14 personas.

Los datos correspondientes al peso son:

77, 58, 89, 55,47,60,54,58,75,65,82,85,75,65

Los correspondientes a la altura:

1.63,1.63,1.85,1.62,1.60,1.63,1.70,1.65,1.78,1.70,1.77,1.83,1.74,1.65

Y los correspondientes a las edades son:

23,23,26,23,26,26,22,23,26,24,28,42,25,26

Realizar las siguientes acciones sobre esa matriz: Seleccionar la primera columna de dos formas diferentes, seleccionar un elemento, seleccionar una fila. Añadir a la matriz la variable sexo, en la última columna de la matriz, que contenga la siguiente información:

"H","M","H","H","M","M","H","M","M","H","H","H","M","M"

Solución Ejercicio 4

Creación de la matriz:

```
> datos= c (77, 58, 89, 55,47,60,54,58,75,65,82,85,75,65,1.63,1.63,1.8  
5,1.62,1.60,1.63,1.70,1.65,1.78,1.70,1.77,1.83,1.74,1.65,23,23,26,23,2  
6,26,22,23,26,24,28,42,25,26)  
> matriz = matrix(datos,14,3,dimnames =list(c(),c("Peso","Altura","Eda  
d")))
```

```
> matriz  
Peso Altura Edad  
[1,] 77 1.63 23  
[2,] 58 1.63 23  
[3,] 89 1.85 26  
[4,] 55 1.62 23  
[5,] 47 1.60 26  
[6,] 60 1.63 26  
[7,] 54 1.70 22  
[8,] 58 1.65 23  
[9,] 75 1.78 26  
[10,] 65 1.70 24  
[11,] 82 1.77 28  
[12,] 85 1.83 42  
[13,] 75 1.74 25  
[14,] 65 1.65 26
```

Seleccionar la primera columna

```
> matriz[,1]  
[1] 77 58 89 55 47 60 54 58 75 65 82 85 75 65  
También
```

```
> matriz[, "Peso"]  
[1] 77 58 89 55 47 60 54 58 75 65 82 85 75 65
```

Seleccionar un elemento

```
> matriz[4, 2]  
Altura  
1.62
```

Seleccionar una fila

```
> matriz[9, ]  
Peso Altura Edad  
75.00 1.78 26.00
```

Añadir a la matriz la variable sexo.

```
> sexo<-factor(c("H","M","H","H","M","M","H","M","M","H","H","H","M","  
M"))  
> sexo  
[1] H M H H M M H M M H H H M M  
Levels: H M  
> matriz<-c(matriz,sexo)  
> matriz
```



```
[1] 77.00 58.00 89.00 55.00 47.00 60.00 54.00 58.00 75.00 65.00 82.00
85.00 75.00 65.00 1.63 1.63 1.85 1.62 1.60 1.63
[21] 1.70 1.65 1.78 1.70 1.77 1.83 1.74 1.65 23.00 23.00 26.00
23.00 26.00 26.00 22.00 23.00 26.00 24.00 28.00 42.00
[41] 25.00 26.00 1.00 2.00 1.00 1.00 2.00 2.00 1.00 2.00 2.00
1.00 1.00 1.00 2.00 2.00
```

```
matriz<-matrix(matriz,14,4,dimnames=list(c(),c("Peso","Altura","Edad",
"Sexo")))
> matriz
```

	Peso	Altura	Edad	Sexo
[1,]	77	1.63	23	1
[2,]	58	1.63	23	2
[3,]	89	1.85	26	1
[4,]	55	1.62	23	1
[5,]	47	1.60	26	2
[6,]	60	1.63	26	2
[7,]	54	1.70	22	1
[8,]	58	1.65	23	2
[9,]	75	1.78	26	2
[10,]	65	1.70	24	1
[11,]	82	1.77	28	1
[12,]	85	1.83	42	1
[13,]	75	1.74	25	2
[14,]	65	1.65	26	2

EJERCICIO 5.

Al añadir a la matriz creada en el Ejercicio 4 la variable sexo R la transforma en tipo numérico asignándole el valor 1 para hombre 2 para mujeres.

Convertir la matriz en un data frame y añadirle una variable de tipo carácter que se corresponda con los nombres de los individuos que son los siguientes:

Juan, Inés, Andrés, Felipe, Pablo, Martina, Germán, Celia, Carmen, Santi, Dani, Antonio, Belinda y Sara

Solución Ejercicio 5

Para añadir la variable de tipo carácter con los nombres de los individuos en el data frame, hacemos lo siguiente:

```
> mode(matriz)
[1] "numeric"
```

Vamos a transformar la matriz en un dataframe:

```
> dataframe<-data.frame(matriz)
> dataframe
```

	Peso	Altura	Edad	Sexo
1	77	1.63	23	1
2	58	1.63	23	2
3	89	1.85	26	1
4	55	1.62	23	1
5	47	1.60	26	2
6	60	1.63	26	2
7	54	1.70	22	1
8	58	1.65	23	2
9	75	1.78	26	2
10	65	1.70	24	1
11	82	1.77	28	1
12	85	1.83	42	1
13	75	1.74	25	2
14	65	1.65	26	2

De esta forma, hemos convertido la matriz de tipo numérico en un conjunto de datos con 4 variables y cada una con 14 observaciones.

Para añadir una variable de tipo carácter, como por ejemplo los nombres de los individuos, en el data frame se puede hacer de la siguiente forma:

```
> Nombres = c ("Juan", "Inés", "Andrés", "Felipe", "Pablo", "Martina",
"Germán", "Celia", "Carmen", "Santi", "Dani", "Antonio", "Belinda", "Sara")
```

```
> Nombres
[1] "Juan"      "Inés"      "Andrés"    "Felipe"    "Pablo"     "Martina"   "Germán"
[2] "Celia"     "Carmen"    "Santi"     "Dani"      "Antonio"
[13] "Belinda"   "Sara"
```

```
> dataframe1 <- data.frame(dataframe, Nombres)
> dataframe1
```

	Peso	Altura	Edad	Sexo	Nombres
1	77	1.63	23	1	Juan
2	58	1.63	23	2	Inés
3	89	1.85	26	1	Andrés
4	55	1.62	23	1	Felipe
5	47	1.60	26	2	Pablo
6	60	1.63	26	2	Martina
7	54	1.70	22	1	Germán
8	58	1.65	23	2	Celia
9	75	1.78	26	2	Carmen
10	65	1.70	24	1	Santi
11	82	1.77	28	1	Dani
12	85	1.83	42	1	Antonio
13	75	1.74	25	2	Belinda
14	65	1.65	26	2	Sara

EJERCICIO 6.

Utilizando el data frame generado en el ejercicio 5 y utilizando la función subset realizar lo siguiente:

- Seleccionar las variables correspondientes a los nombres y su correspondiente sexo.
- Seleccionar las variables correspondientes al peso y a la altura.
- Seleccionar los hombres con altura mayor que 1.70, peso mayor que 65

Solución Ejercicio 6

En el ejercicio 5 habíamos creado la siguiente estructura:

```
> datos= c (77, 58, 89, 55,47,60,54,58,75,65,82,85,75,65,1.63,1.63,1.8
5,1.62,1.60,1.63,1.70,1.65,1.78,1.70,1.77,1.83,1.74,1.65,23,23,26,23,2
6,26,22,23,26,24,28,42,25,26)
> matriz = matrix(datos,14,3,dimnames =list(c(),c("Peso","Altura","Eda
d")))
> matriz
      Peso Altura Edad
[1,]   77   1.63   23
[2,]   58   1.63   23
[3,]   89   1.85   26
[4,]   55   1.62   23
[5,]   47   1.60   26
[6,]   60   1.63   26
[7,]   54   1.70   22
[8,]   58   1.65   23
[9,]   75   1.78   26
[10,]  65   1.70   24
[11,]  82   1.77   28
[12,]  85   1.83   42
[13,]  75   1.74   25
[14,]  65   1.65   26
> sexo<-factor(c("H","M","H","H","M","M","H","M","M","H","H","H","M","
M"))
> matriz<-c(matriz,sexo)
> matriz<-matrix(matriz,14,4,dimnames=list(c(),c("Peso","Altura","Edad
","Sexo")))
> matriz
      Peso Altura Edad Sexo
[1,]   77   1.63   23    1
[2,]   58   1.63   23    2
[3,]   89   1.85   26    1
[4,]   55   1.62   23    1
[5,]   47   1.60   26    2
[6,]   60   1.63   26    2
[7,]   54   1.70   22    1
[8,]   58   1.65   23    2
[9,]   75   1.78   26    2
[10,]  65   1.70   24    1
[11,]  82   1.77   28    1
[12,]  85   1.83   42    1
```

```
[13,] 75 1.74 25 2
[14,] 65 1.65 26 2
> Nombres = c("Juan", "Inés", "Jimena", "Eugenia", "Pablo", "Martina",
,"Germán","Celia","Carmen","Santi","Dani","Antonio","Belinda","Sara")
```

```
> Nombres
[1] "Juan"      "Inés"      "Andrés"    "Felipe"    "Pablo"     "Martina"   "Germá
n" "Celia"     "Carmen"    "Santi"     "Dani"      "Antonio"
[13] "Belinda"   "Sara"
```

```
> dataframe1 <- data.frame(dataframe, Nombres)
```

```
> dataframe1
  Peso Altura Edad Sexo Nombres
1   77   1.63  23    1    Juan
2   58   1.63  23    2    Inés
3   89   1.85  26    1  Andrés
4   55   1.62  23    1  Felipe
5   47   1.60  26    2    Pablo
6   60   1.63  26    2  Martina
7   54   1.70  22    1  Germán
8   58   1.65  23    2    Celia
9   75   1.78  26    2  Carmen
10  65   1.70  24    1    Santi
11  82   1.77  28    1    Dani
12  85   1.83  42    1  Antonio
13  75   1.74  25    2  Belinda
14  65   1.65  26    2    Sara
```

Selección de las variables correspondientes a los nombres y su correspondiente sexo:

```
> subset(dataframe1,select=c(Nombres,Sexo))
```

```
  Nombres Sexo
1    Juan    1
2   Inés    2
3  Andrés    1
4  Felipe    1
5   Pablo    2
6  Martina    2
7  Germán    1
8   Celia    2
9   Carmen    2
10   Santi    1
11   Dani    1
12 Antonio    1
13 Belinda    2
14   Sara    2
```

Selección de las variables correspondientes al peso y a la altura:

```
> subset(dataframe1, select = c(Peso,Altura))
```

```
  Peso Altura
1   77   1.63
2   58   1.63
3   89   1.85
4   55   1.62
5   47   1.60
6   60   1.63
```

```
7    54    1.70
8    58    1.65
9    75    1.78
10   65    1.70
11   82    1.77
12   85    1.83
13   75    1.74
14   65    1.65
```

Selección de los hombres con altura mayor que 1.70, peso mayor que 65:

```
subset(dataframe1, Altura > 1.70 & Peso > 65 & Sexo=="1")
  Peso Altura Edad Sexo Nombres
3    89   1.85  26    1  Andrés
11   82   1.77  28    1   Dani
12   85   1.83  42    1 Antonio
*****
```

EJERCICIO 7.

Escribir una función que devuelva el valor lógico TRUE si la suma de los elementos del vector es menor que 100 y FALSE en caso contrario. Realizar dos llamadas a la función, una con un vector cuyas componentes son los primeros 50 números enteros y otra con un vector cuyas componentes son los primeros 10 números enteros.

Solución Ejercicio 7

En el ejercicio 5 habíamos creado la siguiente estructura:

- ```
> ejemplo = function(x){
 suma = sum(x)
 resultado = ifelse(suma < 100, TRUE, FALSE)
 return(resultado)
}
```
- ```
> x = c(1:50)
> y = c(1:10)
```
- ```
> ejemplo(x)
[1] FALSE
```
- ```
> ejemplo(y)
[1] TRUE
*****
```

EJERCICIO 8.

Instalar el package `ggplot2` siguiendo los pasos explicados en la sección.

EJERCICIO 9.

Importar desde la siguiente URL un fichero .csv:

<https://data.montgomerycountymd.gov/api/views/6rqk-pdub/rows.csv?accessType=DOWNLOAD>

y obtener un summary de la variable Gender.

Solución Ejercicio 9

En el enunciado de este ejercicio se pide acceder a una URL de la que han retirado hace unos días los datos. En lugar de acceder a la URL pedida en el enunciado vamos a acceder a la siguiente:

<http://winterolympicsmedals.com/medals.csv>

File->Import Data Set -> From Text (readr)... Se introduce URL (ver Figura 9-1) y se da a **Update**.

Import Text Data

File/URL: Update

Data Preview:

Year (double)	City (character)	Sport (character)	Discipline (character)	NOC (character)	Event (character)	Event gender (character)	Medal (character)
1924	Chamonix	Skating	Figure skating	AUT	individual	M	Silver
1924	Chamonix	Skating	Figure skating	AUT	individual	W	Gold
1924	Chamonix	Skating	Figure skating	AUT	pairs	X	Gold
1924	Chamonix	Bobsleigh	Bobsleigh	BEL	four-man	M	Bronze
1924	Chamonix	Ice Hockey	Ice Hockey	CAN	ice hockey	M	Gold
1924	Chamonix	Biathlon	Biathlon	FIN	military patrol	M	Silver
1924	Chamonix	Skating	Figure skating	FIN	pairs	X	Silver

Previewing first 50 entries.

Import Options:

Name: ☒ First Row as Names Delimiter: Escape:

Skip: ☒ Trim Spaces Quotes: Comment:

☒ Open Data Viewer Locale: NA:

Code Preview:

```
library(readr)
medals <- read_csv("http://winterolympicsmedals.com/medals.csv")
View(medals)
```

Import Cancel

Figura 9-1: Importar datos CSV desde URL

Podemos, por ejemplo, cambiar el tipo de datos de Event gender de character a factor. Haciendo clic sobre character (situado debajo de Event gender), accedemos a la pantalla de la Figura 9-2 en la que escribimos F, M (los dos valores a contemplar en el factor).

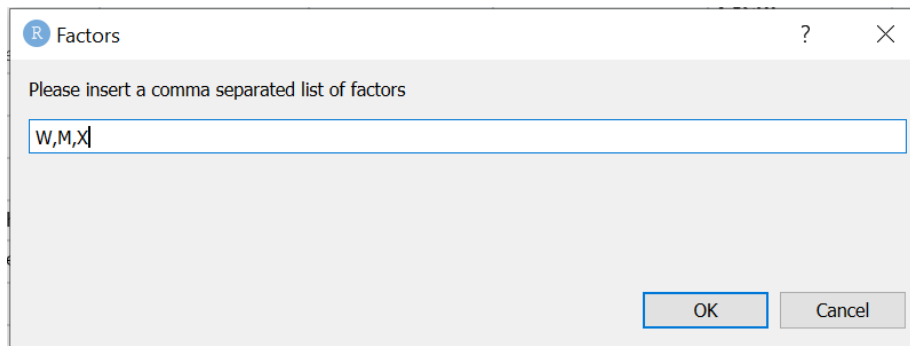


Figura 9-2: Cambio de tipo de datos

y se selecciona **Import**.

	Year	City	Sport	Discipline	NOC	Event	Event gender	Medal
1	1924	Chamonix	Skating	Figure skating	AUT	individual	M	Silver
2	1924	Chamonix	Skating	Figure skating	AUT	individual	W	Gold
3	1924	Chamonix	Skating	Figure skating	AUT	pairs	X	Gold
4	1924	Chamonix	Bobsleigh	Bobsleigh	BEL	four-man	M	Bronze
5	1924	Chamonix	Ice Hockey	Ice Hockey	CAN	ice hockey	M	Gold
6	1924	Chamonix	Biathlon	Biathlon	FIN	military patrol	M	Silver
7	1924	Chamonix	Skating	Figure skating	FIN	pairs	X	Silver
8	1924	Chamonix	Skating	Speed skating	FIN	10000m	M	Gold
9	1924	Chamonix	Skating	Speed skating	FIN	10000m	M	Silver
10	1924	Chamonix	Skating	Speed skating	FIN	1500m	M	Gold
11	1924	Chamonix	Skating	Speed skating	FIN	5000m	M	Gold

Figura 9-3: Parte del contenido del fichero .CSV- data frame medals

En la consola aparece el siguiente código:

```
library(readr)
medals <- read_csv("http://winterolympicsmedals.com/medals.csv",
+   col_types = cols(`Event gender` = col_factor(levels = c("W",
+   "M", "X"))))
> view(medals)
```

Para tener acceso al fichero tenemos que hacer un attach. Por ejemplo, podríamos hacer lo

```
> attach(medals)
summary(`Event gender`)
      W      M      X
802 1386  123
```