
Práctica de introducción a R y R Commander. Aprendiendo lenguaje R desde R Commander

SUPUESTO 1. Elaboración de scripts de R e informes RMarkdown.

Un script es un documento que tiene un conjunto de órdenes o códigos, en este caso para ejecutar en R. En el documento “Entorno de trabajo R Commander.pdf” se recoge la posibilidad de trabajar en R desde R Console, desde un IDE como RStudio o desde una GUI como R Commander (Rcmdr). El objetivo de este ejercicio es elaborar un pequeño script de R trabajando desde R Commander.

- a) Elige el directorio de trabajo para la sesión de prácticas. Una buena costumbre cuando se trabaja en R es comenzar seleccionando el directorio de trabajo. Desde R Commander activa el menú

Fichero ->Cambiar directorio de trabajo.

Elige, por ejemplo, el escritorio y observa que este paso puede ejecutarse incluyendo entrecomillada la ruta del directorio de trabajo elegido como argumento de la función `setwd()`. Cuando sea necesario saber en qué directorio se está trabajando habrá que ejecutar la función `getwd()`.

- a) En la ventana RScript de R Commander (Rcmdr), genera un objeto de R, `x`, con 20 datos de una variable con distribución normal de media 8 y desviación típica 2. Emplea la función `rnorm()` según se indica en los siguientes pasos.

a.1) Una de las formas de obtener ayuda en R sobre la función anterior es ejecutar el código `?rnorm`. Observa como la ayuda, por defecto en html, muestra que esta función tiene tres argumentos: `n`, `mean` y `sd`, esto es, el número de datos que deseamos obtener (`n`), la media (`mean`) y la desviación típica (`sd`) de la normal.

a.2) Usa la función `rnorm()` para obtener los 20 datos la variable con distribución normal. Copia y ejecuta la línea de código:

```
rnorm(20,8,2)
```

Verás que los datos aparecen en la ventana Salida, por debajo de la ventana RScript. El resultado aparece junto con el código que lo ha generado. La ventana RScript muestra sólo el código R.

a.3) Para que sea posible seguir trabajando con los datos que hemos generado, es necesario crear un objeto de R, dando un nombre al resultado de la función `rnorm()`. Como comando de asignación de nombres puede utilizarse el símbolo `<-`. Copia y ejecuta la línea de código:

```
x<-rnorm(20,8,2)#el resultado será el objeto x
```

Observa que R ejecuta la línea de código hasta que aparece el símbolo `#`, empleado para indicar que lo que figura a continuación del símbolo ya no es código R, sino un comentario adicional. Para que R muestre el resultado hay que pedir a R que muestre el objeto `x`, por ejemplo, ejecutando sólo `x`.

Nota: todos los elementos que maneja R son *objetos*: un valor numérico es un objeto, un vector es un objeto, una función es un objeto, una base de datos es un objeto, un gráfico es un objeto, ... R utiliza, de hecho, **programación orientada a objetos**. Ello significa que una misma función hace cosas distintas según la clase del objeto que recibe como argumento.

b) El menú

Distribuciones -> Distribuciones discretas -> Distribución binomial -> Probabilidades binomiales se emplea para obtener la función de masa de probabilidad para una v.a. $X \sim B(n = 10, p = 0.5)$. Observa como este menú permite ejecutar la función **dbinom()**, en este caso como argumento de otras funciones de R.

c) Copia y ejecuta la siguiente línea de código

```
dbinom(0:10, size=10, prob=0.5)
```

observa que el resultado coincide con el resultado del apartado anterior, pero tiene una estructura diferente. El resultado de esta instrucción es un **vector numérico** y en el apartado b) se ha convertido en un **data frame**. Las filas de un data frame representan casos, individuos u observaciones, mientras que las columnas representan atributos, rasgos o variables. Observa como las instrucciones que se han generado desde el menú anterior construyen un data frame al que R Commander por defecto denomina `.Table`. Este data frame está formado por una columna de nombre `Probability`.

d) El menú

Distribuciones -> Distribuciones discretas -> Distribución binomial -> Gráfica de la distribución binomial

se emplea para representar la función de masa de probabilidad para una v.a. $X \sim B(n = 10, p = 0.5)$.

Observa como este menú permite ejecutar la función **plotDistr()**, que proporciona un resultado gráfico. A diferencia de los resultados numéricos de los anteriores apartados, los resultados gráficos se muestra en otra ubicación, en una nueva ventana R Graphics.

e) Guarda el script de instrucciones (`nombre.R`) y los resultados (`nombre.txt`)

e.1) En R la forma habitual de incluir un comentario es añadirlo al código R que se ejecuta. Los comentarios irán precedidos del símbolo `#`. Trabajando con R Commander esto se hará en la ventana R Script.

Al finalizar la sesión, deberás guardar el archivo de instrucciones (script de R) con el menú

Fichero-> Guardar las instrucciones como...

Se trata de un fichero de texto que no incluye los resultados, sólo recogerá las instrucciones y los comentarios añadidos. Este fichero se guarda con extensión `.R` y se podrá revisar por ejemplo con el bloc de notas.

e.2) Observa que R Commander tiene el menú

Fichero-> Guardar los resultados como...

Este menú permite guardar todo lo que aparezca en el bloque "Salida" de la ventana R Commander. Este fichero se guarda con extensión `.txt` y se podrá revisar por ejemplo con el bloc de notas. Comprueba que este archivo de resultados no incluye los resultados gráficos.

f) Guarda el fichero R Markdown (`nombre.Rmd`) y genera el informe (`nombre.html`)

La opción de introducir el comentario en la pestaña R Markdown, es la más indicada cuando se opta por generar un informe html, word o pdf que incluya el código R, los resultados y los comentarios. El comentario debe colocarse fuera de "trozos" ("chunk") que se generan con las instrucciones en R.

Desde la pestaña RMarkdown genera un informe con los enunciados de los apartados b, c y d, las instrucciones generadas y los resultados de estos apartados. Añade algún comentario sobre los resultados. Observa que el informe es un html cuyo nombre por defecto es "RcmdrMarkdown.html". para disponer de informes con otro nombre, es conveniente dar un nombre al fichero RMarkdown desde el menú

Fichero-> Guardar el fichero R Markdown como...

SUPUESTO 2. Exploración de un data frame.

Observa que la barra de herramientas de R Commander no tiene cargado ningún archivo de datos. Para hacer uso de los menús de RCommander, Estadísticos, Gráficos, Modelos, es necesario disponer de un archivo de datos (data frame).

Un **data frame** es una estructura de datos de dos dimensiones (estructura rectangular) como las matrices. Los data frames son una estructura más flexible que una matriz, en una matriz todas las celdas deben contener datos del mismo tipo, mientras que las filas de un data frame admiten datos de distinto tipo (numérico, carácter, lógico, ...). Las filas de un data frame representan casos, individuos u observaciones, mientras que las columnas representan atributos, rasgos o variables. Por tanto, en un data frame se recogen las variables (características) en estudio para un conjunto de casos o individuos que conforman las filas.

Para contribuir al diseño de técnicas de señalización de realidad virtual que resulten naturales y eficaces para la selección de objetos lejanos, se realizó un experimento cuyo objetivo es encontrar los rasgos que mejor describen la forma natural de apuntar a objetivos lejanos.

Se recogieron datos de movimiento de puntos de la mano, el brazo y el cuerpo, mientras los participantes apuntan a 27 objetivos en una sala virtual.

La descripción del experimento puede verse en el siguiente enlace y en el documento **VrPointing.pdf** incluido en el campus virtual.

<https://github.com/TorSalve/pointing-data-ATHCC/blob/master/README.md>

Dalsgaard, TorSalve et al (2021). Modeling Pointing for 3D Target Selection in VR. *VRST '21*:

Proceedings of the 27th ACM Symposium on Virtual Reality Software and Technology.

<https://doi.org/10.1145/3489849.3489853>.

En el trabajo se recopilaban datos que describen el movimiento de los participantes mientras señalan. La información está repartida en cuatro archivos que incluyen la información de los participantes, datos de calibración, datos de los movimientos corporales de los participantes y datos de los objetivos a los que apuntaron.

El archivo **vrPart.RData** describe a los participantes en el estudio. Incluye observaciones de las siguientes variables:

- `pid`: ID del participante
- `gender`: sexo del participante
- `age` - edad del participante
- `forearmLength` - longitud del antebrazo
- `forearmMarkerDist` - distancia del marcador del antebrazo al codo
- `indexFingerLength` - longitud del dedo índice
- `upperArmLength` - longitud de la parte superior del brazo
- `upperArmMarkerDist` - distancia desde el marcador de la parte superior del brazo hasta el codo
- `height` - altura del participante
- `rightShoulderMarkerDist.X` distancia horizontal desde el marcador del hombro derecho hasta el hombro del participante
- `rightShoulderMarkerDist.Y` - distancia vertical desde el marcador del hombro derecho hasta el hombro del participante

Se empleará este archivo de datos para responder con R Commander a las siguientes cuestiones sobre exploración de un data frame, incluyendo además la obtención de la información sobre los participantes presentada en el apartado 2.1 del trabajo **VrPointing.pdf**.

- a) Descarga el archivo de datos **vrPart.RData** desde el campus virtual al directorio de trabajo. Desde R Commander (Rcmdr), carga el archivo de datos **vrPart.RData**.

Datos-> Cargar Conjunto de datos...

Observa que se ha empleado la función **load()**, que la barra de herramientas tiene cargado el archivo de datos y que se puede visualizar.

Observa la pestaña R Markdown y añade sobre ella los comentarios que consideres oportunos sobre la realización con R Commander de los distintos apartados.

- b) Usa el menú

Datos-> Conjunto de datos activo->Variables del conjunto de datos activo...

para que R muestre los nombres de las variables de este conjunto de datos. Observa que este menú simplemente ejecuta la instrucción **names(vrPart)** para volcar los nombres de las variables del data frame. Observa cómo se organizan las variables en el archivo de datos.

- c) ¿Qué clasificación hace R de las variables en estudio? Utiliza la instrucción **str(vrPart)**. Al comenzar a trabajar con un data frame es recomendable revisar cuál es la estructura del archivo de datos en uso y para ello puede emplearse la función **str()**. Desde R Commander puede ejecutarse código R escribiéndolo en la ventana R Script. Prueba a ejecutar la instrucción **str(vrPart)** para ver la estructura de este archivo de datos.

Recuerda la clasificación de las variables estadísticas (características) en cualitativas y cuantitativas, verás que las variables cualitativas o categóricas para R son **variables factor** y las modalidades son los niveles del factor (**levels**). Si el resultado de la función **str()** mostrará sólo algunos de los niveles del factor. Se pueden presentar todos los niveles del factor con la instrucción **levels(vrPart\$gender)**.

- d) Emplea el menú

Estadísticos->Resúmenes->Conjunto de datos activo...

para observar la estadística descriptiva básica que proporciona la función **summary()** aplicada al conjunto de datos. ¿Qué resultados proporciona para cada tipo de variables?

En el data frame hay una columna que incluye el código asignado a cada participante en el estudio. ¿Tienen sentido los cálculos de la función **summary()** para esta variable?

Observa que desde el menú Datos se pueden hacer cambios sobre el archivo de datos. Para cambiar el tipo de variable:

Datos-> Modificar variables del conjunto de datos activo->Convertir variable numérica en factor...

El menú empleado hace uso de la función **as.factor()**.

- e) Compara la salida del apartado anterior con la proporcionada por el menú

Estadísticos-> Resúmenes-> Resúmenes numéricos...

para las variables **age** (edad), **height** (altura del participante), **forearmLength** (longitud del antebrazo) e **indexFingerLength** (longitud del dedo índice). Anota la media y la desviación típica de la edad y de la longitud del antebrazo de los participantes (resultados presentados en el apartado 2.1 del trabajo **VrPointing.pdf**.)

- f) En los apartados anteriores se hace uso de las funciones **summary()**, aplicada a todo el conjunto de datos, y **numSummary()**, aplicada a algunas variables numéricas. Emplea las instrucciones **?summary** y **?numSummary** para acceder a la ayuda de cada función y comprobar que son funciones incluidas en diferentes paquetes de R, la primera es del paquete **base** y la segunda de uno de los paquetes que se instalan con R Commander, **RcmdrMisc**. Observa la cabecera del html de ayuda: **summary {base}**, **numSummary {RcmdrMisc}**. La forma de vincular una función de R con el paquete del que proviene es incluir el nombre del paquete entre llaves.

- g) Representa en un gráfico de barras la distribución de los participantes según el género. Observa que los resultados gráficos emplean una ventana diferente a las otras salidas de resultados.
- h) En los apartados anteriores se emplearon las funciones `as.factor()`, `Barplot()` y `numSummary()` para obtener resultados descriptivos. Revisa el código R generado por estas funciones, así como los resultados de la función `str()`, para comprobar que las variables de un data frame se identifican con el **nombre de la variable, ligado siempre al nombre del archivo de datos**. Identifica tres formas diferentes de seleccionar los datos de la variable `height`. Prueba a utilizar las diferentes opciones aplicando la función `summary()` a los datos de esta variable.
- i) Utiliza el menú

Estadísticos->Resúmenes->Resúmenes numéricos...

para obtener, por separado según el género, la estadística descriptiva básica de las variables cuantitativas de este archivo de datos.

Observa que es necesario disponer de una variable para formar grupos con los que segmentar el archivo de datos y que **la(s) variable(s) que forma(n) los grupos son variables tipo factor**.

En este caso, figura entre los resultados, **NA**, esta es la forma de denotar en R los valores perdidos (NA: not available).

- j) Algunas funciones, como la función `numSummary()`, admiten un **argumento groups**, que debe ser una variable factor, para presentar los resultados segmentando el archivo de datos según los niveles de un factor.

Otra forma de calcular alguno de los parámetros descriptivos básicos por separado para cada grupo es utilizar el **símbolo ~**. Prueba a utilizar el menú

Estadísticos → Resúmenes → Tabla de estadísticas...

Para obtener, por separado según el género, la mediana de las variables numéricas del archivo de datos. Con este menú se ejecuta el siguiente código R.

```
Tapply(age ~ gender, median, na.action=na.omit, data=vrPart) # median by groups
```

- k) En la instrucción anterior, R emplea la expresión **age ~ gender** para indicar que se quiere trabajar con la variable `age` por separado para cada uno de los niveles del factor `gender`. El **símbolo ~** también se puede emplear entre variables cuantitativas. En caso de emplearlo entre variables cuantitativas, se estará indicando que se busca definir la relación la variable a la izquierda (variable dependiente) del símbolo `~` y la(s) variable(s) independiente(s) colocadas a la derecha del símbolo `~`.

Comprueba esto revisando el código R que se genera desde el menú

Gráficas→Diagrama de dispersión

para representar la relación entre la altura (`height`, variable x) y la longitud del dedo índice (`indexFingerLength`, variable y). En este caso, el primer argumento de la función es **`indexFingerLength~height`**.

- l) Al terminar la sesión de trabajo podemos guardar:
- el archivo de datos (archivo.RData*),
 - * estamos identificando los archivos RData con ficheros de datos, pero realmente son archivos específicos de R que pueden almacenar diferentes objetos (varios archivos de datos, varios objetos creados con funciones R, ...) en un único fichero .RData.
 - el script de trabajo (archivo.R),
 - el archivo de resultados (archivo.txt) y
 - generar el informe de prácticas R Markdown o guardar el archivo R Markdown (archivo.Rmd) para generar el informe en otro momento.

l.1) Guarda el archivo de instrucciones (script) `solTema0.R`. Observa que se trata de un archivo de texto que se puede abrir con el bloc de notas.

1.2) Guarda el archivo de resultados solTema0.txt. Observa que se trata de un archivo de texto que se puede abrir con el bloc de notas.

1.3) Genera el informe R Markdown. Puedes guardar previamente el archivo R Markdown con el nombre solTema0.Rmd, para que quede en el directorio de trabajo un archivo html con el nombre solTema0.html con los resultados y comentarios del supuesto 2.

1.4) Sube los tres archivos al enlace habilitado en el curso virtual.

R Commander. Resumen de menús empleados.
--

Fichero → Cambiar directorio de trabajo Datos → Cargar conjunto de datos... Barra de herramientas de Rcmdr (Visualizar conjunto de datos) Datos → Conjunto de datos activo → Variables del conjunto de datos activo Estadísticos → Resúmenes → Conjunto de datos activo Uso de las funciones str(), names() Estadísticos → Resúmenes → Resúmenes numéricos Gráficas → Grafica de barras... Estadísticos → Resúmenes → Resúmenes numéricos (resúmenes por grupos) Gráficas → Diagrama de dispersión Fichero → Guardar las instrucciones como... Fichero → Guardar los resultados como... Datos → Conjunto de datos activo → Guardar el conjunto de datos activo Uso de # para incluir anotaciones en el script Objetos en R (variables)
--