PROYECTO DE TITULACIÓN

Roberto Bustillos Huilca, MVZ, MSc.

Universidad Nacional de Loja

Carrera de Medicina Veterinaria y Zootecnia

Décimo ciclo, período octubre 2018 - marzo 2019

Contenido

1. El entorno R. Motivos para su uso.
2. Distribución e instalación de R.
3. Inicio de las sesiones en R.
4. R como calculadora científica.
5. La ayuda en R.
6. Bibliografía de R.
7. Primeras nociones: comandos, funciones y objetos.
8. Permanencia de los datos y eliminación de objetos.
9. Edición y ejecución de comandos desde ficheros editables: scripts.
10. Paquetes (bibliotecas o librerías) de R.

El entorno R. Motivos para su uso.

- R es, básicamente, un lenguaje que permite implementar técnicas estadísticas. Es a la vez:
  - un entorno interactivo para el análisis estadístico y gráfico,
  - y un lenguaje de programación interpretado de alto nivel con funciones orientadas a objetos.
- El diseño de R vino muy influenciado por dos lenguajes existentes: S (Becker, Chambers and Wilks) y Scheme (Sussan).
- El resultado es un lenguaje de apariencia similar a S, pero en el que la implementación y semántica subyacente se derivan de Scheme.

El entorno R. Motivos para su uso.

- ¿Por qué debería cambiar a R cuando ya tengo bastante conocimiento de otros programas estadísticos?
- Si únicamente vamos a utilizar unos cuantos test estadísticos, y no pensamos hacer nada más en el futuro, quizá no valga la pena el esfuerzo. PERO hay muchos motivos para aprender su uso...

El entorno R. Motivos para su uso.

- Flexibilidad: R está concebido como entorno de programación con multitud de comandos y funciones específicas en estadística que permite fácilmente implementar y evaluar técnicas nuevas.
- Sacar provecho de su gran cobertura (no tiene rival) y la gran disponibilidad de aplicaciones de vanguardia en infinidad de campos.
- Entender la literatura más actual. Cada vez más gente presenta sus resultados en el contexto de R. La mayoría de los "popes" de cada área ya se ha cambiado. Aún más: CONTRIBUYEN y COMPARTEN.
- La distribución de R viene acompañada de un numeroso conjunto de funciones (librerías base). Sin embargo, existen a libre disposición numerosas librerías específicas con todas las últimas técnicas disponibles (y además con explicación de su uso).

El entorno R. Motivos para su uso.

- El hecho de que R sea un lenguaje de programación podría desanimar a muchos usuarios que piensan que no tienen "alma de programadores". Nada más lejos de la realidad:
  - R es un lenguaje interpretado, es decir, los comandos escritos en el teclado son ejecutados directamente sin necesidad de construir ejecutables.
  - Además la sintaxis es simple e intuitiva.
- La estructura y facilidad de uso de R nos permite implementar nuestras propias funciones y rutinas a medida que aparecen nuestras necesidades.
- Además, también puede utilizarse para realizar gráficos de alta calidad de enorme utilidad en los trabajos de investigación.
- Y por encima de todo, es GRATIS. Uno de los mejores softwares integrados del mundo y resulta que es nuestro por nada.

Distribución e instalación de R.

- Para instalar R hay que bajar el fichero ejecutable de la página web del proyecto R: https://www.r-project.org/
- En esta página debemos hacer click en download CRAN, escoger a continuación uno de los servidores (CRAN Mirrors) y después seguir los pasos según el sistema operativo.
- Actualmente (julio 2018), la versión más reciente de R es la 3.5.1.

Tarea

Instalar la última versión de R tras desinstalar cualquier versión previa que pudiera haber en el equipo de trabajo.

Return

Inicio de las sesiones en R.

- Abriendo R, por defecto, se abre una sola ventana, consola o ventana de comandos de R en la cual introduciremos comandos y será dónde se verán los resultados de los análisis.
- Justo después de la cabecera, aparece una línea en blanco con el símbolo $>$ en el margen izquierdo: PROMPT.
- A partir de aquí R espera que escribamos COMANDOS e instrucciones para comenzar a trabajar.
- Para ejecutar un comando escrito utilizamos la tecla ENTER.
- El signo # indica la introducción de un comentario.
- Las órdenes elementales consisten en expresiones o en asignaciones:
  - Si una orden consiste en una expresión, se evalúa, se imprime y su valor se pierde.
  - Una asignación, por el contrario, evalúa una expresión, no la imprime y guarda su valor en una variable.

Inicio de las sesiones en R.

- En cada línea sólo caben 128 caracteres: si queremos escribir más, podemos utilizar otra línea que comienza con el símbolo $+$ en el lado izquierdo de la pantalla en vez de $>$. Esto también puede ocurrir cuando el comando que terminamos de escribir no está completo (falta un paréntesis, etc.).
- Si encontramos que el error es la falta de algo podemos teclearlo, y al pulsar ENTER, ejecutará lo que hayamos solicitado.
- Con la tecla $\uparrow$ recuperamos las instrucciones de la sesión y con las teclas $\rightarrow$ $\leftarrow$ podemos corregir errores.
- Para separar expresiones se emplea ; (punto y coma).
- q ( ) es el comando para salir de R.

R como calculadora científica.

- El uso más básico de R sería utilizarlo como una calculadora científica. Así: 3+4
  Sumará 3 y 4 y devolverá el resultado.
- Cualquier función matemática que se nos ocurra está en R:
  log(x), exp(x), log(x,n), log10(x), sqrt(x), factorial(x), choose(n,x), gamma(x), lgamma(x), floor(x), round(x,digits=0), signif(x,digits=6), cos(x), sin(x), tan(x), abs(x).
  Y podemos aplicarla sobre cualquier número.

R como calculadora científica.

- Sin embargo, lo que convierte a R en una potente herramienta de trabajo es que está diseñado de forma que la mayoría de operaciones (p.e. +, -, /, etc.) y de funciones (p.e. log( )) están definidas con caracter vectorial, es decir operar componente a componente.
- La función principal para definir un vector es a través de sus componentes, con la función c ( ), mediante el comando más importante de R que es <- el de la asignación.
- Los paréntesis ( ) se emplean para los argumentos de las funciones y para agrupar expresiones algebraicas. Los corchetes [ ] o dobles corchetes [[ ]] para seleccionar partes de un objeto así como el $. Las llaves { } para agrupar expresiones.
- Para referirnos a la componente enésima del vector v escribimos v[n].

Tarea

Ejecutar los siguientes comandos:

- log(((3+2)*5)+6)
  Equivale a log({(3+2)*5}+6) o log({{3+2}*5}+6)
- v <- c(1,2,3,4)
  Crea un vector llamado v con cuatro valores.
- v
  Nos muestra el valor de dicho vector.
- w <- c(1,-2,2,4)
  Crea un vector llamado w.
- w[2]
  Nos muestra la segunda componente de w.

Tarea

Calcular las siguientes operaciones, analizar los resultados obtenidos y observar los mensajes de advertencia:

- 2*v-3*w+2
  Las operaciones afectan a todas las componentes.
- v*w
  Multiplica componente a componente.
- w/v
  Divide componente a componente.
- v^2
  Eleva al cuadrado cada componente.
- v^w
  Eleva cada componente de v a su correspondiente componente de w.

Tarea

- sqrt(w)
  Produce un warning por pedir que haga la raíz de un negativo.
- vw <- c(v,w)
  Une los dos vectores y los almacena en uno nuevo vw.
- vw
  Muestra el resultado de vw.
- vwa <- c(vw,6)
  Une otra componente.
- vwa/v
  Da un warning pues ambos vectores no tienen el mismo número de componentes. Pero, ¿qué hace?

La ayuda en R.

- R dispone de una ayuda muy completa sobre todas las funciones, procedimientos y elementos que configuran el lenguaje.
- Además de las opciones de menú propias de R, desde la ventana de comandos se puede acceder a información específica sobre las funciones de R con el comando help o mediante ?.
  - help(log), nos abre la ventana de ayuda sobre la función log.
  - ? log, hace lo mismo.

Tarea

Consultar la ayuda sobre los comandos glm( ) y boxplot( ).
Consultar la ayuda sobre if (observar la diferencia entre ?if, help("if") y help(if).

La ayuda en R.

- Cuando no sepamos el nombre exacto del comando o de la función que necesitamos, pero sí que sepamos el tema sobre el que queremos ayuda (en inglés), podemos utilizar la función help.search( ).
- Por ejemplo, si queremos introducir datos, y no sabemos que comandos nos pueden ayudar:
  - help.search("data imput"), con un poco de suerte aparecerá información sobre el nombre de funciones asociadas a esa búsqueda.
  - ?read.table, nos aportará dicha información.
- Es posible acceder a manuales vía el menú Ayuda-Manuales (en PDF) o vía CRAN: https://www.r-project.org/

Bibliografía de R.

- La bibligrafía de R es extensísima! Imposible de darla toda entera.
- En la página web del proyecto existe disponible mucha Documentation que además es de libre distribución.
- Entre la enorme cantidad de libros (posiblemente en la actualidad haya más de 100!) enteramente dedicados a R destacamos dos:
  - John M. Chambers (2008). "Software for Data Analysis: programming with R". Springer, New York.
  - Peter Delgaard (2008). "Introductory Statistics with R". 2nd edition. Springer.
- Una muy buena referencia es: Michael J. Crawley (2007). The R book. John Wiley and Sons, Ltd.

Primeras nociones: comandos, funciones y objetos.

- R es mucho más que una calculadora científica en la que aplicar comandos o instrucciones.
- R es un lenguaje orientado a objetos. Bajo este término se esconde la simplicidad y flexibilidad de R.
- Mientras que programas más clásicos muestran directamente los resultados de un análisis. R almacena los resultados en objetos, para ser observados o analizados posteriormente, produciendo unas salidas mínimas.
- Esto puede ser un poco extraño para el usuario, pero esta característica suele ser muy útil. De hecho, el usuario puede extraer sólo aquella parte de los resultados que le interesa.

Clases de objetos.

- Los vectores son el tipo básico de objeto en R, pero existen más tipos.
- Las matrices o, más generalmente, variables indexadas (Arrays) son generalizaciones multidimensionales de los vectores. De hecho, son vectores indexados por dos o más índices.
- Los factores sirven para representar datos categóricos.
- Las listas son una forma generalizada de vector en las cuales los elementos no tienen por qué ser del mismo tipo y a menudo son a su vez vectores o listas. Las listas permiten devolver los resultados de los cálculos estadísticos de un modo conveniente.
- Las hojas de datos o data frames son estructuras similares a una matriz, en que cada columna puede ser de un tipo distinto a las otras.
- Las funciones son también objetos de R que pueden

Examples

- Vectores
  v <- c(5,4,5,4,5); enfer <-
  c("Aftosa","Malaria","Dengue","Zika","Chagas")
- Listas
  lista <- list(Vaca = "Magda", Toro = "José",enfermos =
  TRUE,número.hijos = 3, edad.hijos = c(4, 6, 8))
- Funciones
  y <- -2; log <- log(8,abs(y))
- Matrices
  a<-1:10;matrix(a,nrow=2,ncol=5); matrix(a,2,5,byrow=T)
- Factores
  sexo <- c("Macho", "Hembra", "Macho", "Macho", "Macho",
  "Hembra", "Hembra"); sexo <- factor(sexo)

Examples

- Data frame
  mtcars; str(mtcars); data <- mtcars
  peso <- c(14.21, 10.36, 11.89, 13.81, 12.03); edad <- c(22, 34, 29, 25, 30); sexo <- c("M","H","H","M","M"); data1 <- data.frame(edad,tiempo,sexo)

Más sobre objetos.

- R distingue entre mayúsculas y minúsculas, de tal modo que A y a son símbolos distintos y se referirán, por tanto, a objetos distintos.
- Los nombres de los objetos pueden contener letras mayúsculas o minúsculas, junto con números y puntos(NO blancos, NO%, No$.)
- Durante una sesión de trabajo con R los objetos que se crean se van almacenando por su nombre.
- La función objects ( ) se puede utilizar para obtener los nombres de los objetos almacenados en R. Equivalente a la función ls ( ).

Permanencia de los datos y eliminación de objetos.

- Ya hemos comentado que la colección de objetos almacenados en cada momento se denomina espacio de trabajo (workspace).
- Los objetos creados durante una sesión de R pueden almacenarse en un archivo para su uso posterior (archivos.RData).
- Es posible eliminarlos con el comando rm( ).

Más sobre permanencia de los datos y eliminación de objetos.

- Como hay nombres que tendemos a repertir, es recomendable utilizar un directorio de trabajo diferente para cada problema que analicemos con R. es posible cambiar de directorio desde el Menú Archivo o con el comando setwd("C:/midir").
- Si no sabemos en qué directorio estamos podemos averiguarlo con el comando getwd( ).
- A veces nos interesa ejecutar varias órdenes que tengamos almacenadas en un archivo, p.e. ordenes.R. Si el fichero está en el directorio de trabajo, es posible ejecutarla dentro de una sesión de R con la orden source("ordenes.R").

Edición y ejecución de comandos desde ficheros editables: scripts.

- Para operaciones que requieran varias instrucciones consecutivas, resulta especialmente útil trabajar con un fichero de comandos editables (script). El mantenimiento del código ordenado y comentado es una "Buena práctica estadística".
- R proporciona por defecto la posibilidad de trabajar con scripts como ventanas del propio programa.
- Podemos abrir un script nuevo desde el menú Archivo/Nuevo script. Desde ellos, podemos ejecutar con CTRL + r los comandos de la línea en la que estamos, o el bloque de comandos que tengamos seleccionado.
- Existen diferentes editores que pueden facilitar el trabajo con R; colores de sintaxis, completa paréntesis, etc.
  - RStudio, TinnR, RKward.
  - Notepad, (X)Emacs.

RStudio

- RStudio es otro editor para scripts disponible para usuarios de R en todos los sistemas operativos.
- Los autores lo consideran un entorno de desarrollo integrado que combina un interfaz muy intuitiva con herramientas de código muy potentes que permiten sacar el máximo provecho a R.
- La versión es gratuita y está disponible en: https://www.rstudio.org/
- Este editor ofrece una serie de opciones no existentes en R, entre otras, por ejemplo, comprobar rápidamente que ningún paréntesis queda sin cerrarse o marcar, copiar y pegar columnas.
- Pero además nos da un listado de las variables y nos da una descripción de los bancos de datos que hemos introducido. También tiene una lista de los paquetes instalados y los gráficos realizados.

Paquetes (bibliotecas o librerías) de R.

- Hasta ahora no hemos mencionado mucho la palabra estadística. Sin embargo R tiene implementado muchas técnicas estadísticas, tanto clásicas como modernas.
- R consta de un sistema base (enorme cantidad de técnicas y numéricas) y de paquetes (packages) adicionales.
- Muchos paquetes están disponibles a través de Internet en CRAN: https://cran.r-project.org/.

Tarea

Conectarse a CRAN y observar los paquetes disponibles (hasta la fecha julio 2018, 12721).

Instalación de paquetes (bibliotecas o librerías) de R.

- Desde el menú Paquetes > Instalar paquete(s)...
- También podemos utilizar la función install.packages( ).
- La instalación no implica que los paquetes ya puedan ser utilizados. Es necesario cargar las librerías antes de empezar a usarlas. Lo mismo ocurre con las librerías existentes en la versión local de R.

Tarea

Instalar el paquete foreign y spdep.

Incorporando las librerías en la sesión de R.

- Dos maneras de hacerlo: la primera mediante el menú Paquetes > Cargar paquete; y la segunda utilizando la función library ( ); library (foreign).
- El comando library ( ) abre una ventana con información sobre las librerías (paquetes) instalados en R.
- La función library y help conjuntamente, nos da información sobre estas librerías.
  library(help="foreign").