

Curso teórico-práctico de estudios de genoma completo en poblaciones mezcladas

Introducción a R

¿Qué es R?


Es un paquete estadístico para la manipulación, cálculo y presentación gráfica de datos

Distribución gratuita (<http://cran.r-project.org/>)


Autores: W. R. Venables, D. M. Smith y un grupo de programadores.

Se puede instalar en Linux, MACOSX y Windows

Referencias

- Introductory Statistics with R, Peter Dalgaard
 - Modern Applied Statistics with S, Bill Venables and Brian Ripley
 - Applied Statistical Genetics with R, Andrea S. Foulkes
- 

Ventajas

- Es gratuito y cada vez más común
 - Es un lenguaje de programación que tiene implementadas una amplia gama de pruebas estadísticas
 - Hay paquetes disponibles con técnicas y métodos para casi a todas las disciplinas y aplicaciones
 - Es una herramienta gráfica particularmente poderosa
 - Tiene muchos recursos de ayuda disponibles y un foro de discusión
- 

Operaciones

Puede servir como una calculadora

`5 * 4`

`exp(-2)`

`sqrt(12)`

`3^5`

Comúnmente queremos almacenar los resultados, para hacer esto asignamos el resultado a una variable.

`a <- (3 * 7) + 1`



Operaciones

Para ver el contenido de una variable, por ejemplo a, simplemente teclee el nombre de la variable y presione “enter”.

Las variables pueden ser manipuladas de manera que nosotros queramos, por ejemplos, podemos elevarla al cuadrado

a^2



Vectores

A menudo necesitamos trabajar con vectores de números. Los vectores se pueden asignar de varias maneras:

Un vector de enteros consecutivos

```
v1 <- 11:15
```

En ejemplo más general, una secuencia de valores que va del 11 y hasta 15, en unidades de 0.5

```
v2 <- seq(11, 15, by = 0.5)
```

También se puede asignar el vector directamente

```
v3 <- c(11, 2, 73, 24, 35)
```



Vectores

Podemos manipular vectores como si fueran números

`v3^2 + 6`

Podemos contar cuantos elementos del vector son mayores o iguales que 13.5

`sum(v3 > 13.5)`

Podemos extraer un elemento específico del vector

`v3[4]`

o los tres primeros elementos

`v3[1:3]`



Vectores

Si queremos los elementos 1, 2 y 5

```
v3[c(1, 2, 5)]
```

Los vectores también pueden ser cadenas de caracteres

```
v4 <- c("Raul", "Victor", "Sandra", "Yolanda", "Silvia")
```



Matrices

Se pueden construir matrices a partir de vectores

```
v1 <- 1:12
```

```
m1 <- matrix(v1, nrow = 3, ncol = 4, byrow = TRUE)
```

```
m2 <- matrix(v1, nrow = 3, ncol = 4)
```

Como los vectores, también las matrices pueden ser manipuladas como si fueran variables

```
m1^2 + 3
```



Matrices

También podemos extraer elementos específicos, por ejemplo, el elemento fila 2, columna 3

`m1[2, 3]`

o la fila 2 completa, la columna 3 completa,

`m1[2,]` o `m1[,3]`



Listas

Es posible almacenar datos de diferentes tipos y tamaños en un objeto llamado listas.

Crea una lista que incluye los vectores v1, v4 y m1

```
l1 <- list(v1 = v1, v4 = v4, m1 = m1)
```

Para extraer los componentes de la lista se pueden utilizar el símbolo \$ o [].
Por ejemplo,

```
l1$v1 o l1[2]
```



Objetos almacenados

- Las variables, vectores, matrices y listas que se han creado se almacenan hasta que cerramos la sesión de R. Para ver todos los objetos almacenados se utiliza el siguiente comando:

`ls()`



Funciones

Hasta ahora se han utilizado las funciones `exp`, `sqrt`, `seq`, `c`, `matrix`, and `ls`.

El formato de un función consiste en el nombre de la función seguida de un paréntesis que contiene los argumentos de la función.

```
matrix(1:4, 2, 2)
```

Podemos crear nuestras propias funciones

```
mi.f <- function(x) { x^2 - 2*x + 4 }
```

```
mi.f(3)
```

La función `curve()` permite graficar una función sobre un rango de valores dado.

```
curve(mi.f(x), from = -10, to = 10)
```



Ayuda y lectura de datos

Fuente de ayuda

```
help.start()
```

R tiene una función muy útil para leer datos externos llamada `read.table()`

```
a <- read.table(file = "xxx.txt", header = TRUE)
```

```
help(read.table)
```

Muchas funciones trabajan con los datos en un formato que se llama “data.frame”, por lo que a veces es necesario convertir los datos ejecutando el siguiente comando,

```
d1 <- data.frame(a)
```

```
attach(d1)
```

```
dim(d1)
```



Gráficas y resúmenes numéricos

Usaremos los datos de cystfibr.txt

```
cystfibr <- read.table(file = "cystfibr.txt", header = TRUE)
```

```
attach(cystfibr)
```

```
tlc
```

```
mean(tlc)
```

```
sd(tlc)
```

```
range(tlc)
```

```
quantile(tlc)
```

```
summary(cystfibr)
```



Gráficas y resúmenes numéricos

```
par(mfrow = c(2, 1))  
hist(tlc[sex == 0], xlim = c(80, 150))  
hist(tlc[sex == 1], xlim = c(80, 150))  
hist(tlc, freq = FALSE)  
lines(density(tlc))  
par(mfrow = c(1, 1))  
boxplot(tlc ~ sex)
```

