FACULTAD DE MATEMÁTICAS
PONTIFICIA UNIVERSIDAD
CATÓLICA DE CHILE

EYP1113 - Probabilidad y Estadística

Laboratorio 01

Pilar Tello Hernández
pitello@uc.cl

Facultad de Matemáticas
Departamento de Estadística
Pontificia Universidad Católica de Chile

Segundo Semestre 2021

Equipo Laboratorio

Profesora:

- Pilar Tello → pitello@uc.cl

Ayudantes:

- Miércoles:
  - Camila Echeverría → cecheverriac@uc.cl
  - Vanesa Reinoso → vcreinoso@uc.cl
- Jueves:
  - Yaku Valdivia → yaku@uc.cl
  - Francisca Vilca → fvilca@uc.cl

Introducción a R

R es un conjunto integrado de programas para manipulación de datos, cálculo y gráficos.

Introducción a R

R es un software estadístico de libre acceso el cual puede ser utilizado en diferentes sistemas operativos como Windows, MacOS y Linux.

R es un lenguaje de programación en el que se introducen códigos para posteriormente ser ejecutados.

Una de las grandes ventajas de R es que es un programa de código abierto en el que miles de personas de todo el mundo colaboran en el desarrollo de nuevas metodologías, de manera que se pueden acceder a los paquetes descargándolos como también compartir los propios paquetes con otros.

Introducción a R

Instalación

La descarga del archivo de instalación de R se realiza desde uno de los links de abajo dependiendo del sistema operativo:

- Microsoft Windows: `http://cran.r-project.org/bin/windows/base/`
- OSX: `http://cran.r-project.org/bin/macosx/`
- Linux: `http://cran.r-project.org/bin/linux/`

Una vez instalado R, se puede instalar R Studio desde:

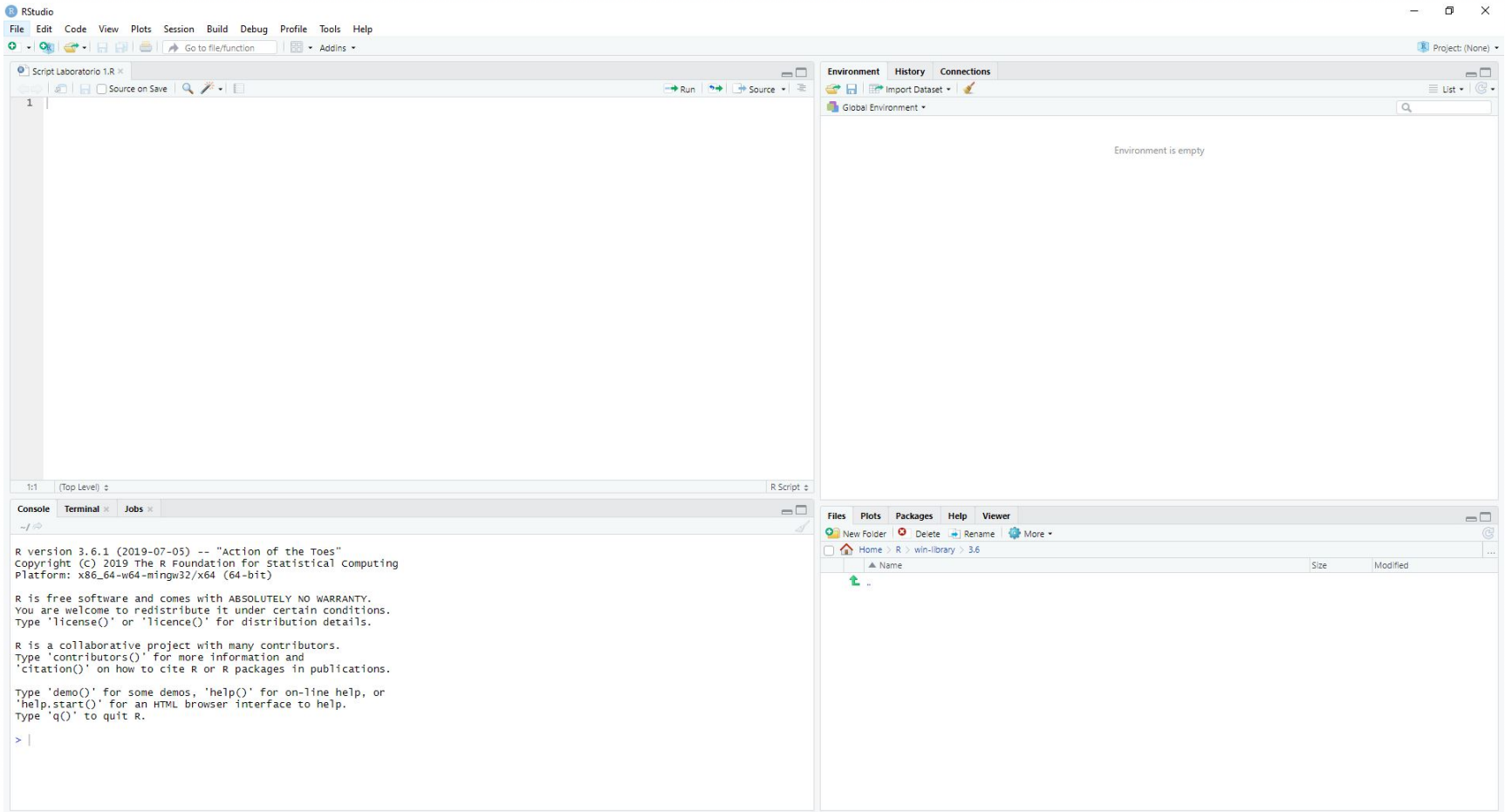
- `https://www.rstudio.com/products/rstudio/download/`

Como alternativa también existe R Studio Cloud:

- `https://rstudio.cloud/`

Introducción a R

Instalación



Introducción a R

En la consola de R, se puede escribir directamente el código de R , y pulsar **enter** para ver el valor. Sin embargo, esta no es la manera más eficiente de trabajar en R.

Si se quiere guardar el trabajo, corregirlo, repetirlo, etc., es más conveniente usar el editor de R. Se debe seleccionar archivo, nuevo Script (documento en blanco del editor) en el cual se puede escribir los programas y guardar.

La ejecución del código desde el script se hace desde cualquier posición de la línea o seleccionando

Windows: Tecla F5 o Control + Enter, MacOS: Cmd + Enter.

Se puede incluir comentarios que R no leerá si utilizamos el símbolo # al comienzo de la línea.

Introducción a R

Operadores en R

En R es posible llevar a cabo distintas operaciones matemáticas y aritméticas usando operadores básicos tales como $+, -, /, **, *$.

Calcule en la consola de R las siguientes operaciones y presione ENTER. Observe el resultado.

- 1+2+3
- 100 + 200
- 3.14 + 2.17
- 2 - 1
- 3/4
- 1/3
- 2*33
- 2^3
- 2**3
- 2*5+200/2^3

Los resultados obtenidos de cualquier operación, o de aplicar una función, van apareciendo antecedidos por el símbolo $[n]$, mientras que cualquier código o sentencia que escribamos aparecerá se destacará con $>$.

Introducción a R

Funciones usadas en R

A continuación se presentan distintos comandos de utilidad usados comúnmente en R.

Funciones matemáticas:

Nombre de la función	Descripción
`sqrt`	Raíz cuadrada
`log, log2, log10`	Logaritmos
`exp`	Función exponencial
`abs`	Valor absoluto
`sign`	Signo
`cos, sin, tan`	Funciones trigonométricas
`acos, asin, atan`	Funciones trigonométricas inversas
`%%`	Resto de una división
`factorial, lfactorial`	Factorial y su logaritmo

Introducción a R

Definición de objetos

`R` es un software que permite la creación de objetos de varios tipos: alfanuméricos, escalares, vectores, matrices, etc. Los valores que se asignen a los objetos quedan guardados en la memoria de `R` mientras dure la sesión de trabajo. Los objetos pueden definirse de la siguiente forma:

```
objeto <- expresión
```

```
objeto = expresión
```

donde `objeto` es el nombre que se le asigna al objeto, y `expresión` puede ser una fórmula, un vector, una palabra, etc.

El código se puede comentar utilizando #.

Introducción a R

Definición de objetos

En R podemos definir distintos tipos de variables, los que conoceremos primero son los siguientes:

- Variables numéricas: se definen los objetos como el número u operación numérica a definir.
  - `a <- 3*6+9/7`
  - `b <- sqrt(9)/log(10)`
- Variables booleanas: en `R` los valores booleanos se definen como `TRUE` o `T` y `FALSE` o `F`.
  - `d <- T`
  - `e <- FALSE`
- Variables de texto: los strings o char se escriben siempre entre comillas .
  - `f <- "Laboratorio EYP1113"`
  - `g <- "Segundo Semestre 2021"`

Introducción a R

Vectores

Si necesitamos obtener la clase o tipo de objeto que tengo guardado en un objeto a. El comando a utilizar es el siguiente: `class(a)`. Esto nos retornará el tipo del objeto creado.

Para crear vectores debemos utilizar el comando `c()`.

- `numeros <- c(1,2,3,4,5)`
- `textos <- c("a","b","c","d","e")`
- `booleanos <- c(T,F,TRUE,FALSE)`

Para nombrar ciertos elementos de un vector, se utiliza el comando `names`.

- `notas <- c(3.5,4.1,5.5,6.0)`
- `nombres <- c("I1","I2","I3","Ex")`
- `names(notas) <- nombres`

Introducción a R

Operaciones con vectores

En R podemos realizar operaciones con vectores, por ejemplo:

- `v1 <- c(1,1,1,3)`
- `v2 <- c(2,0,2,1)`
- `v1+v2`
- `v1-v2`
- `v1*v2`
- `v1/v2`
- `v1*2`
- `v1/2`

También podremos crear funciones, pero R, viene con funciones o comandos ya creados para operar con vectores o para crearlos tales como:

- `sum(v1)`
- `prod(v1)`
- `mean(v1)`
- `sd(v1)`
- `min(v1)`
- `max(v1)`

Pueden encontrar una lista más completa de funciones de R en: `https://cran.r-project.org/doc/contrib/Short-refcard.pdf`

Introducción a R

Expresiones lógicas

Podemos realizar comparaciones lógicas, por ejemplo:

- `1>2`
- `1<2`
- `1>=2`
- `1<=2`
- `1==2`
- `1!=2`

Comparaciones con los vectores creados anteriormente:

- `v1<2`
- `v1==1`
- `v1<v2`
- `v1>=v2`

Para unir condiciones lógicas el "y" se representa con carácter `&` y el "o" con el carácter `|`.

- `v2<2 & v2>0`
- `v2<2 | v2>0`

Introducción a R

Manipulación de vectores

Trabajaremos con los vectores `numeros` y `textos`. Para conocer el largo de un vector está el comando `length()`. Para poder acceder al `i`-ésimo componente del vector `v` debemos ejecutar `v[i]`.

- `length(nombres)`
- `length(textos)`
- `nombres[1]`
- `textos[2]`

Para acceder a más de un índice, entonces debemos entregar un vector dentro de los corchetes de la forma `v[c(i,j,k,...)]`

- `nombres[c(1,2)]`
- `textos[c(2,3)]`

Un modo de crear una secuencia desde un índice `i` hasta un índice `j` es con el comando `i:j`.

- `1:10`
- `5:10`

Introducción a R

Manipulación de vectores

Con lo visto anteriormente podemos acceder a varios términos continuos dentro de un vector de mayor longitud. Crearemos un vector de mayor longitud para dar un ejemplo.

- `v3 <- 1:20`
- `v3[5:15]`

Retomando el uso de `names()` y la manipulación de vectores podemos utilizar:

- `notas[2]`
- `notas["I2"]`
- `notas[c("I3","I1","Ex")]`
- `notas[notas>4]`

Podemos guardar distintos filtros lógicos en variables auxiliares tales como:

- `notas>4`
- `notas<6`
- `notas==max(notas)`
- `notas==min(notas)`
- `notas>mean(notas)`
- `notas<=mean(notas)`

Introducción a R

Manipulación de vectores

Para manipular vectores con variables categóricas es conveniente convertir nuestra variable a `factor`. En el caso de ser nominal, no importa el orden.

- `textos2 <- factor(textos)`

En el caso en que nuestra variable sea ordinal, con el comando `factor()` podemos indicar el orden de los niveles de la variable con el argumento `levels=`.

- `opiniones <- c("Bueno", "Malo", "Neutro", "Bueno","Malo", "Malo", "Neutro", "Neutro")`
- `opiniones <- factor(opiniones)`
- `opiniones <- factor(opiniones, levels=c("Malo", "Neutro", "Bueno"))`

Introducción a R

Ayuda en R

En R podemos buscar ayuda para funciones con los `help` o `?`

- `help(class)`
- `?class`

Siempre se puede buscar ayuda en Google para problemas que tengan con sus códigos. Stack Overflow es un sitio recomendable para realizar consultas.

Introducción a R

Matrices

En R podemos definir una matriz con el comando `matrix()`. Si al comando matrix le entregamos la secuencia guardada en `v3`.

- `m <- matrix(v3)`

Por defecto, nos crea una matriz de tamaño 20×1. Nosotros podemos entregarle la cantidad de filas y/o columnas que queremos que tenga la matriz con los argumentos `nrow=` y `ncol=`. También podemos indicarle si queremos o no que la matriz se rellene por filas, con el argumento `byrow=` que puede tomar los valores `TRUE` o `FALSE`. Podemos ver un ejemplo creando la matriz de 4 filas y 5 columnas. Rellenando la matriz por filas.

- `m1 <- matrix(v3,nrow=4,ncol=5,byrow=TRUE)`

Introducción a R

Matrices

Podemos llamar a un elemento o a un conjunto de datos mediante de una matriz `m` llamando a `m[i,j]`, donde `i` y `j` puede ser un elemento o un vector de posiciones de fila o columna. Si se deja el espacio de `i` o `j` en blanco, `R` entiende que se está llamando a todas las filas o columnas. Podemos ver ejemplos ejecutando lo siguiente:

- `m1[1,]`
- `m1[,1]`
- `m1[1,1]`
- `m1[c(1,2),c(3,4)]`

También podemos seleccionar la matriz sin las filas o columnas indicadas anteponiendo un signo "-" al número o vector indicado:

- `m1[-1,]`
- `m1[,-1]`
- `m1[-1,-1]`
- `m1[-c(1,2),-c(3,4)]`

Introducción a R

Matrices

Tal como en los vectores podemos asignar nombres a las filas y columnas, esta vez con los comandos `rownames()` y `colnames()`.

- `nombresfilas <- c("f1","f2","f3","f4")`
- `nombrescolumnas <- c("c1","c2","c3","c4","c5")`
- `rownames(m1) <- nombresfilas`
- `colnames(m1) <- nombrescolumnas`

Introducción a R

Operaciones con Matrices

Para realizar operaciones simples con matrices tales como:

- `m1*3`
- `m1/3`
- `m1+3`
- `m1**3`
- `m1+m1`
- `m1-2*m1`

Comparaciones lógicas:

- `m1>5`
- `m1<=5`
- `m1>5 & m1<15`
- `m1>15 | m1<5`

Con esto podemos filtrar la matriz:

- `m1[m1>5]`
- `m1[m1<=5]`
- `m1[m1>5 & m1<15]`
- `m1[m1>15 | m1<5]`

Introducción a R

Funciones asociadas a matrices

Algunas funciones asociadas a matrices en `R`:

Función	Descripción
`diag()`	Diagonal
`*`	Producto elemento a elemento
`%*%`	Producto matricial
`dim()`	Dimensiones
`ncol()`	Número de columnas
`nrow()`	Número de filas
`t()`	Transpuesta
`det()`	Determinante
`solve()`	Inversa
`rowSums()`	Suma de filas
`colSums()`	Suma de columnas
`rowMeans()`	Promedio simple de filas
`colMeans()`	Promedio simple de columnas

Introducción a R

Funciones asociadas a matrices

Podemos unir filas o columnas a una matriz con los comandos `rbind()` y `cbind()`

- `nuevafila <- c(5,10,15,20,25)`
- `m2 <- rbind(m1,nuevafila)`
- `nuevacolumna <- c(3,6,9,12,15)`
- `m3 <- cbind(m2,nuevacolumna)`