FACULTAD DE MATEMÁTICAS
PONTIFICIA UNIVERSIDAD
CATÓLICA DE CHILE

EYP1113 - Probabilidad y Estadística

Laboratorio 02

Pilar Tello Hernández
pitello@uc.cl

Facultad de Matemáticas
Departamento de Estadística
Pontificia Universidad Católica de Chile

Segundo Semestre 2021

Introducción a R

Bases de datos

En `R` se pueden abrir distintos tipos de bases de datos. También hay varias bases que vienen integradas como `mtcars`. Pueden revisar un listado más completo con `data()`. Antes de aprender a abrir una base de datos vamos a aprender algunas funciones utiles. Las funciones `head()` y `tail()` nos permiten ver las primeras y últimas 6 observaciones de una base respectivamente.

- `head(mtcars)`
- `tail(mtcars)`

El comando `str()` nos entrega una breve descripción de la base de datos y el tipo de variables que contiene.

- `str(mtcars)`

La función `summary()` nos entrega estadísticas descriptivas de las variables de la base. En el caso de las variables numéricas nos entrega: mínimo, primer cuartil, mediana, media, tercer cuartil y máximo. En el caso de variables categóricas realiza un conteo de éstas.

- `summary(mtcars)`

Introducción a R

Bases de datos

Para crear nuestra propia base de datos podemos utilizar la función `data.frame()`
Con el ejemplo anterior de las notas vamos a crear una nueva base.

- `Tipo <- c("I1","I2","Ex")`
- `Nota <- c(3.5,6.0,5.5)`
- `Azul <- c(F,T,T)`
- `Libreta <- data.frame(Tipo,Nota,Azul)`
- `str(Libreta)`
- `summary(Libreta)`

Esta función reconoce inmediatamente los nombres de los vectores como nombres de las columnas y las filas las indexa con el número de la observación. Los nombres de las columnas también se pueden definir al momento de crear la base de datos de la siguiente forma:

- `Libreta2 <- data.frame(Tipo2=Tipo,Nota2=Nota,Azul2=Azul)`

Introducción a R

Bases de datos

Podemos acceder a los subconjuntos de esta base de datos tal como se hacía con la matrices. También podemos llamar a la columna que requiramos por su nombre.

- `Libreta[c(1,3),c("Azul","Nota")]`

Si requerimos acceder a la columna `v1` de una base de datos `df`, también se pueden usar las siguientes notaciones:

- `df$v1`
- `df["v1"]`
- `Libreta$Nota`
- `Libreta["Nota"]`

Introducción a R

Bases de datos

Para acceder a un subconjunto de datos con alguna restricción en específico existe la función `subset()` a la cual entregamos la base de datos como primer argumento y luego como segundo argumento `subset=` se entregan las condiciones requeridas.

- `subset(Libreta,subset=Azul==TRUE)`
- `subset(Libreta,subset=Nota>5)`

Otra forma de filtrar que es frecuentemente usada es mediante los corchetes, indicando condiciones para filas y también seleccionando columnas de interés.

- `Libreta[Libreta$Azul==TRUE,]`
- `Libreta[Libreta$Azul==TRUE & Libreta$Nota>5,]`

Como se puede ver, la diferencia radica en que con la función `subset` se llama directamente a las variables por su nombre, porque se entrega la base completa en la función, en cambio al realizar el filtro "manual" se debe llamar a la base y con el signo `$` indicar recién el nombre de la variable.

Introducción a R

Bases de datos

Para ordenar una base de datos en función de una variable existe la función `order()`. Por ejemplo para ordenar la base de datos en orden creciente de `Nota`. Si se antepone el signo "-" entonces se ordena de manera decreciente.

- `Libreta[order(Libreta$Nota),]`
- `Libreta[order(-Libreta$Nota),]`

Para agregar una nueva fila a una base de datos del tipo `data.frame` se puede utilizar el comando `rbind()` conocido anteriormente, siempre preocupándonos que la o las nuevas filas tengan los mismos nombres de columnas de la base de datos.

- `nuevafila <- data.frame(Tipo="I4", Nota=4.5, Azul=TRUE)`
- `nuevaLibreta <- rbind(Libreta, nuevafila)`

Para agregar una nueva columna se puede escribir dentro de la misma base.

- `nuevaLibreta$nuevacolumna <- nuevaLibreta$Nota+1`

Introducción a R

Bases de datos

Los nombres de filas y columnas se pueden renombrar con los comandos `rownames()` y `colnames()`. También se puede renombrar una o más filas o columnas en específico indicando la posición de la fila o columna dentro de los corchetes que hemos ocupado anteriormente.

- `colnames(nuevaLibreta) <- c("Tipo2", "Nota2", "Azul2", "Columna2")`
- `colnames(nuevaLibreta)[4] <- "NuevaColumna2"`

Introducción a R

Listas

Hasta ahora ya hemos visto que podemos crear vectores, matrices y bases de datos (`data.frame`). Para juntar todos estos tipos de variables y guardarlos en un objeto existen las listas. con el comando `list()` se pueden crear listas con distintos tipos de objetos en su interior. Sólo le debemos entregar los objetos que se quieren almacenar en ella.

- `lista <- list(opiniones,m1,Libreta)`

Tal como en los `data.frame`, podemos asignarle nombre a los componentes de la lista.

- `lista2 <- list(v=opiniones,  m=m1, bd=Libreta)`

Luego podemos acceder a estos objetos llamándolos por su nombre establecido o por su coordenada dentro de la lista:

- `lista[1]`
- `lista2$v`

Introducción a R

Instalación de paquetes

R contiene dos tipos de paquetes, los del tipo Base los cuales están incorporados automáticamente en la instalación de R, y los paquetes de contribución los cuales se deben descargar para su instalación.

Ejecutando el comando getOpcion() en la consola se obtiene las aplicaciones que contiene el paquete base

```
getOption("defaultPackages")
"datasets"  "utils"     "grDevices" "graphics"  "stats"     "methods"
library(help = "datasets")
```

Existe una gran variedad de paquetes de contribución que son aporte de personas a lo largo del mundo (los cuales son gratuitos). Se requiere conexión a internet para descargarlo e instalarlo y se debe ejecutar

```
install.packages("Nombre")
```

Una vez instalado el paquete se carga con el comando

```
library(Nombre)
```

Introducción a R

Lectura de bases de datos

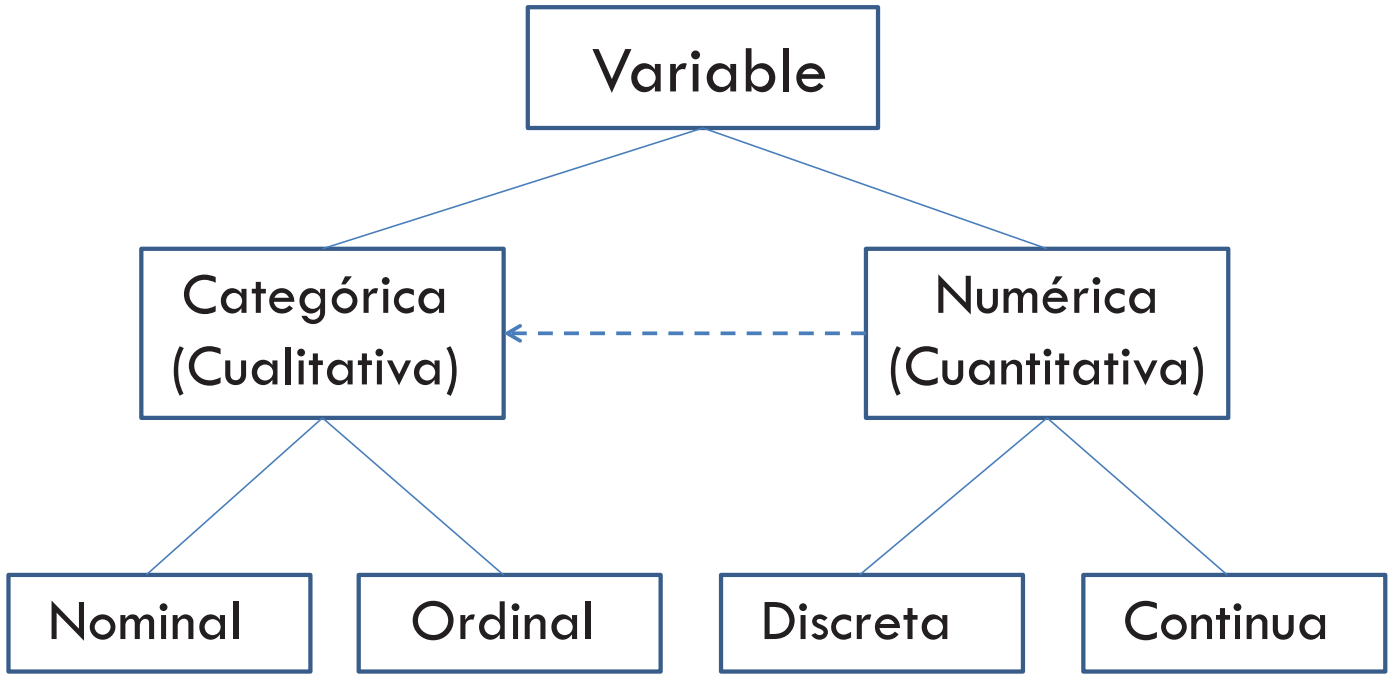
Antes de comenzar cualquier análisis de datos es importante saber:

- Formato en que se encuentra la información: TXT, DAT, XLS, XLSX, CSV, SPSS, SAS, SQL, ACCES, etc.
- Indicador de dato faltante: “na”, “NA”, “,”, “;”, “ ”, etc.
- Números especiales: ”88”, ”888”, ”99”, ”999”, etc.

En resumen, tener el llamado libro de variables a mano.

Introducción a R

Lectura de bases de datos



Introducción a R

Lectura de bases de datos

La mayoría de las veces la información que se recolecta se presenta de la siguiente manera:

	Variables						
Observación	X_1	X_2	X_3	$\cdots$	X_j	$\cdots$	X_K
1	$x_{1\,1}$	$x_{1\,2}$	$x_{1\,3}$	$\cdots$	$x_{1\,j}$	$\cdots$	$x_{1\,K}$
2	$x_{2\,1}$	$x_{2\,2}$	$x_{2\,3}$	$\cdots$	$x_{2\,j}$	$\cdots$	$x_{2\,K}$
3	$x_{3\,1}$	$x_{3\,2}$	$x_{3\,3}$	$\cdots$	$x_{3\,j}$	$\cdots$	$x_{3\,K}$
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\cdots$	$\vdots$	$\vdots$	$\vdots$
i	$x_{i\,1}$	$x_{i\,2}$	$x_{i\,3}$	$\cdots$	$x_{i\,j}$	$\cdots$	$x_{i\,K}$
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\cdots$	$\vdots$	$\vdots$	$\vdots$
N	$x_{N\,1}$	$x_{N\,2}$	$x_{N\,3}$	$\cdots$	$x_{N\,j}$	$\cdots$	$x_{N\,K}$

donde $x_{i\,j}$ corresponde a los valores (o nombres) de toman las variables para una i-ésima observación.

Introducción a R

Lectura de bases de datos

Para importar bases de datos en formato TXT, DAT y Excel, en la consola de `R` se utilizan las siguiente funciones:

- `read.table()`: importa bases de datos en formato TXT, DAT y CSV.
- `read.csv()`: importa bases de datos en formato CSV.
- `read_excel()`: importa bases de datos en formato XLS y XLSX (paquete `readxl o tidyverse`).
- `scan()`: importa un vector de datos.
- `import()`: importa distintos tipos de datos (paquete `rio`). Para ver el listado de tipos de datos completo: `https://cran.r-project.org/web/packages/rio/vignettes/rio.html`.

Introducción a R

Lectura de bases de datos

Las condiciones climáticas son muy importantes para practicar deportes. La base de datos `Tenis.txt` contiene información diaria que se ha recolectado sobre condiciones climáticas y la decisión final de un jugador profesional de Tenis para practicar el deporte:

Variable	Descripción
`Dia`	identificador del día evaluado
`Pronostico`	pronóstico del día: Soleado, Nublado o Lluvioso
`Temperatura`	temperatura pronosticada del día: Calido, Moderado o Frio
`Temperatura Maxima`	temperatura máxima pronosticada en grados Celsius
`Temperatura Minima`	temperatura mínima pronosticada en grados Celsius
`Humedad`	humedad pronosticada del día: Alta o Normal
`Viento`	intensidad del viento pronosticada del día: Alta o Debil
`Juega_Tenis`	indica si el jugador juega finalmente tenis o no. 1: Sí 0: No

Introducción a R

Lectura de bases de datos

Una vez que los datos ya están leídos, a veces es de utilidad usar el comando `attach()`, el cual permitirá trabajar directamente con ellos. Además, con la función `names()` se podrán obtener los nombres de las variables correspondientes.

Introducción a R

Lectura de bases de datos

Las dos formas más comunes de leer una base de datos son:

- `data <- import(file.choose())`
- `data <- import("..../Tenis.txt")`

donde la función `file.choose()` permite seleccionar directamente un archivo de la unidad de trabajo, sea cual sea la extensión.

Introducción a R

Lectura de bases de datos

En caso donde la lectura es través del directorio, una sentencia muy conveniente es:

- `getwd()`
- `setwd()`

Para obtener la dirección del directorio que necesitamos se puede utilizar la función `choose.dir()`, esto abrirá una ventana en la que debemos buscar en el computador el directorio y luego escogerlo, así obtendremos la dirección deseada. Para Mac existe la función `choose_dir()` de la librería `easycsv`.

Ejemplo:

- `choose.dir()`
- `setwd("../EYP1113/2021 - 02/Laboratorio/Laboratorio 02/")`
- `data <- read.table("Tenis.txt", header=TRUE)`
- `data <- import("Tenis.txt")`

Introducción a R

Lectura de bases de datos

Se puede obtener la clase de cada columna de la base junto con una pequeña estadística descriptiva mediante la sentencia:

- `str()`

Ejemplo:

- str(data)

Introducción a R

Lectura de bases de datos

Observemos que la `Temperatura_Maxima` se ha leído como un factor, es decir, una variable categórica. Para corregir este error al abrir la base de datos debemos indicarle a la función `read.table` que en nuestra base de datos los decimales vienen delimitados con una coma (,) con el argumento `dec=","`.
Ejemplo:

```
data <- read.table(file.choose(),header=TRUE,dec=",")
data <- import(file.choose(),dec=",")
```

Introducción a R

Lectura de bases de datos

La variable `Juega_Tenis` se ha considerado una variable numérica, cuando debería ser una variable categórica. Esto se puede modificar con la sentencia:

- `as.factor()`

En el caso contrario se podría modificar con la sentencia:

- `as.numeric()`

Ejemplo:

- `as.factor(data$Juega_Tenis)`