DataFrames

**Introducción**

Los data frames (marcos de datos) son el objeto más habitual para almacenar datos en R. Es parecido a las matrices con la diferencia es que las matrices almacenan tipos de datos **homogéneos** mientras que los data frames almacenan tipos de datos **heterogéneos**.

Como ejemplo, suponer que se tiene los siguientes datos:



Sin embargo, el resultado no es satisfactorio, ya que todas las variables se han transformado a la clase carácter. En cambio, si se usa data.frame mantendrá el tipo original de las variables.

**Operaciones**

* **Creación de un dataframe: data.frame()**







* **Mostrar cabecera**

names(datos)

* **Mostrar los valores de columnas**

#Seleccionar mes

datos$mes

datos[, 1] # Equivalente

# Seleccionando las columnas 1 a 3 con una secuencia

datos[, 1:3]

# Seleccionando columnas con la función 'c'

datos[, c("temperatura", "precipitaciones")]

datos[, c(2, 4)] # Equivalente

# Observación de la primera fila y segunda columna

datos[1, 2]

# Primera y segunda fila de la segunda columna

datos[1:2, 2]

* **Acceso directo utilizando la función attach**

Para poder acceder directamente sobre las columnas del dataframe y no tenerlo que escribir una y otra vez:

attach(datos)

\* En este caso las columnas del dataframe podrían estar presentes en el entorno global (.GlobalEnv) y esto podría causar conflictos de nombres

* **Deshabilitar el acceso directo**

detach(datos)

* **Añadir columnas y filas**

El dataset cars contiene 50 filas y 2 variables. Recargar:

data(cars)

head(cars)

Suponer que se quiere crear una nueva variable para transformar la velocidad en kilómetros por hora (km/h) y la distancia en metros:

kilómetro = milla/0.62137

metro = pie/3.2808



Agregar dos nuevas columnas llamadas kph (kilómetros por hora) y meters (metros) con el siguiente código:

cars$kph <- cars$speed / 0.62137

cars$meters <- cars$dist / 3.2808

También se puede hacer uso de la función **cbind**. Para agregar nuevas filas usar usar la función **rbind**.

kph <- cars$speed / 0.62137

meters <- cars$dist / 3.2808

cars <- cbind(cars[, c(1, 2)], kph, meters)

* **Eliminar variables o filas**

Para eliminar columnas se utiliza el singo meno (-)

# Borrando la primera y segunda columna con el signo -

cars2 <- cars[, -c(1, 2)]

# Seleccionar solo las columnas que queremos conservar

cars2 <- cars[, c("kph", "meters")]

# Comprobar

head(cars2)

* **Ordenar data frames**

Usar el data.frame de R **mtcars** y la función **order**:

data(mtcars)

# **Índice** de ordenación en base a la variable 'hp'

ii <- order(mtcars$hp)

# Solo mostramos las primeras 4 columnas

head(mtcars[ii, 1:4])



También se puede ordenar de mayor a menor haciendo uso del signo menos.

ii <- order(-mtcars$hp)

head(mtcars[ii,])

 

* **Establecer diferentes condiciones para ordenar los datos**

ii <- order(mtcars$cyl, mtcars$hp)

head(mtcars[ii, 1:4])



* **Filtrar data frames**

Uso de la función **subset** para crear subconjuntos según los valores de alguna columna.

Por ejemplo, visualizar el subconjunto de datos donde el número de cilindros del automóvil es = **6 y** la potencia en caballos es > **115**.

subset(mtcars, cyl == 6 & hp > 115)



Seleccionar los coches donde la variable wt es menor que 2 o la variable hp  es mayor a 115.

subset(mtcars, wt < 2 | hp > 115)

