

# Clase Data Frames

Adaptada de Nadia Testani

# Data Frames

- ▶ Un data frame es una **lista cuyos componentes pueden ser vectores o matrices**, cuyas longitudes (o numeros de filas en el caso de las matrices) deben coincidir en todos los componentes.
- ▶ Los elementos que componen el dataframe no son **necesariamente del mismo tipo** (aunque si tienen la misma longitud)
- ▶ En general, un data frame se puede pensar como una tabla, es por esto que **solo puede tener dimension 2**.

# Genero Data Frame

Sintaxis:

```
data.frame(nombre_columna_1 = vector_1,  
nombre_columna_2 = vector_2, ...)
```

*O*

```
data.frame(nombre_columna_1 = vector_1,  
matriz, ...)
```

*Obs: el largo del vector debe coincidir con la cantidad de filas de la matriz*

### *Ejemplo a partir de vectores*

```
rm(list = ls())  
  
n <- c(2, 3, 5)  
s <- c("aa", "bb", "cc")  
b <- c(TRUE, FALSE, TRUE)  
df <- data.frame(n, s, b)  
df
```

```
##   n  s    b  
## 1 2 aa  TRUE  
## 2 3 bb FALSE  
## 3 5 cc  TRUE
```

## Ejemplo a partir de matriz y vector

```
rm(list = ls())
```

```
nombres <- c("Padre", "Madre", "Hijo")
```

```
m <- matrix(c(1:6),nrow=3)
```

```
v <- c(180,157,165)
```

```
df <- data.frame(familia = nombres, m, estatura=v)  
df
```

```
##   familia X1 X2 estatura  
## 1  Padre  1  4      180  
## 2  Madre  2  5      157  
## 3  Hijo   3  6      165
```

```
mi_df <- data.frame(  
  "entero" = 1:4,  
  "factor" = c("a", "b", "c", "d"),  
  "numero" = c(1.2, 3.4, 4.5, 5.6),  
  "cadena" = as.character(c("a", "b", "c", "d"))  
)
```

## Podemos usar `dim()` en un data frame

```
dim(mi_df)
```

```
## [1] 4 4
```

*# El largo de un data frame es igual a su número de columnas*

```
length(mi_df)
```

```
## [1] 4
```

*# names() nos permite ver los nombres de las columnas*

```
names(mi_df)
```

```
## [1] "entero" "factor" "numero" "cadena"
```

Si los vectores que usamos para construir el data frame no son del mismo largo, los datos no se reciclarán. Se nos devolverá un error.

```
# data.frame(  
#   "entero" = 1:3,  
#   "factor" = c("a", "b", "c", "d"),  
#   "numero" = c(1.2, 3.4, 4.5, 5.6),  
#   "cadena" = as.character(c("a", "b", "c", "d"))  
# )
```

```
matriz <- matrix(1:12, ncol = 4)
```

```
#Usamos as.data.frame() para coercionar una matriz a un da
```

```
df <- as.data.frame(matriz)
```

```
#Verificamos el resultado
```

```
class(df)
```

```
## [1] "data.frame"
```

```
#mi_df * 2
```

# Manipulación de Data Frames

*Cargamos dataframe de ejemplo de R*

```
df <- mtcars
```

## Cantidad de filas

*nrow(dataframe)*

## Ejemplo

```
nrow(df)
```

```
## [1] 32
```

## Cantidad de columnas

*ncol(dataframe)*

## Ejemplo

```
ncol(df)
```

```
## [1] 11
```



## Veo las primeras filas

*head(dataframe)*

### Ejemplo

```
head(df)
```

##	mpg	cyl	disp	hp	drat	wt	qsec	vs	am	gear	carb
## Mazda RX4	21.0	6	160	110	3.90	2.620	16.46	0	1	4	4
## Mazda RX4 Wag	21.0	6	160	110	3.90	2.875	17.02	0	1	4	4
## Datsun 710	22.8	4	108	93	3.85	2.320	18.61	1	1	4	1
## Hornet 4 Drive	21.4	6	258	110	3.08	3.215	19.44	1	0	3	1
## Hornet Sportabout	18.7	8	360	175	3.15	3.440	17.02	0	0	3	2
## Valiant	18.1	6	225	105	2.76	3.460	20.22	1	0	3	1

*Muestra las primeras 6 filas*

# Acceder a elementos del data frame e indexación

## Acceder a un elemento

### 1. Por numero de filas y columnas:

Sintaxis:

```
dataframe[numero_fila, numero_columna]
```

### Ejemplo

```
df[1,4]
```

```
## [1] 110
```

```
class(df[1,4])
```

```
## [1] "numeric"
```

## 2. Por nombre de filas y columnas:

Sintaxis:

```
dataframe["nombre_fila", "nombre_columna"]
```

### Ejemplo

```
df["Mazda RX4", "hp"]
```

```
## [1] 110
```

```
class(df["Mazda RX4", "hp"])
```

```
## [1] "numeric"
```

## Acceder a una columna como data frame

### 3. Por numero de la columna:

Sintaxis:

```
dataframe[numero_columna]
```

o

```
dataframe[c(numero_columna_1,  
numero_columna_2)]
```

*Esto ultimo es para acceder a varias columnas*

## Ejemplo

```
selecciono_columna_1 <- df[1]  
head(selecciono_columna_1)
```

```
##                mpg  
## Mazda RX4      21.0  
## Mazda RX4 Wag  21.0  
## Datsun 710      22.8  
## Hornet 4 Drive  21.4  
## Hornet Sportabout 18.7  
## Valiant        18.1
```

```
class(df[1])
```

```
## [1] "data.frame"
```

## 4. Por nombre de la columna:

Sintaxis:

```
dataframe["nombre_columna"]
```

O

```
dataframe[c("nombre_columna_1",  
"nombre_columna_2")]
```

*Esto ultimo es para acceder a varias columnas #*

## Ejemplo

```
selecciono_columna_mpg <- df["mpg"]  
head(selecciono_columna_mpg)
```

```
##           mpg  
## Mazda RX4    21.0  
## Mazda RX4 Wag 21.0  
## Datsun 710    22.8  
## Hornet 4 Drive 21.4  
## Hornet Sportabout 18.7  
## Valiant      18.1
```

```
class(df["mpg"])
```

```
## [1] "data.frame"
```

## Acceder al contenido de una columna

### 1. Por numero de la columna:

Sintaxis:

```
dataframe[[numero_columna]]
```

### Ejemplo

```
df[[1]]
```

```
## [1] 21.0 21.0 22.8 21.4 18.7 18.1 14.3 24.4 22.8 19.2 17.8 16.4 17.3 15.2 10.4  
## [16] 10.4 14.7 32.4 30.4 33.9 21.5 15.5 15.2 13.3 19.2 27.3 26.0 30.4 15.8 19.7  
## [31] 15.0 21.4
```

```
class(df[[1]])
```

```
## [1] "numeric"
```

## 2. Por nombre de la columna:

Sintaxis:

```
dataframe[["nombre_columna"]]
```

O

```
dataframe$nombre_columna
```

## Ejemplo

```
df[["mpg"]]
```

```
## [1] 21.0 21.0 22.8 21.4 18.7 18.1 14.3 24.4 22.8 19.2 17.8 16.4 17.3 15.2 10.4  
## [16] 10.4 14.7 32.4 30.4 33.9 21.5 15.5 15.2 13.3 19.2 27.3 26.0 30.4 15.8 19.7  
## [31] 15.0 21.4
```

```
class(df[["mpg"]])
```

```
## [1] "numeric"
```



## Acceder a una fila como data frame

### 1. Por numero de la fila:

Sintaxis:

```
dataframe[numero_fila,]
```

O

```
dataframe[c(numero_fila1, numero_fila_2),]
```

*Esto último es para acceder a mas de una fila*

## Ejemplo

```
df[2,]
```

```
##           mpg cyl disp  hp drat   wt  qsec vs am gear carb
## Mazda RX4 Wag  21   6  160 110  3.9 2.875 17.02  0  1    4    4
```

```
class(df[2,])
```

```
## [1] "data.frame"
```

## 2. Por nombre de la fila:

Sintaxis:

```
dataframe["nombre_fila",]
```

O

```
dataframe[c("nombre_fila_1",  
"nombre_fila_2"),]
```

*Esto último es para acceder a mas de una fila*

## Ejemplo

```
df["Datsun 710",]
```

```
##           mpg cyl disp  hp drat   wt  qsec vs am gear carb  
## Datsun 710 22.8   4  108  93 3.85 2.32 18.61  1  1   4     1
```

```
class(df["Datsun 710",])
```

```
## [1] "data.frame"
```

# Indexación lógica

Sintaxis

```
dataframe[condicional sobre  
dataframe$nombre_columna, ]
```

O

```
subset(dataframe, condicional sobre  
nombre_columna)
```

## Ejemplo

```
df[df$mpg > 20,] #me quedo con las filas en las que se verifique que los valores de la columna "mpg" sean .
```

	mpg	cyl	disp	hp	drat	wt	qsec	vs	am	gear	carb
## Mazda RX4	21.0	6	160.0	110	3.90	2.620	16.46	0	1	4	4
## Mazda RX4 Wag	21.0	6	160.0	110	3.90	2.875	17.02	0	1	4	4
## Datsun 710	22.8	4	108.0	93	3.85	2.320	18.61	1	1	4	1
## Hornet 4 Drive	21.4	6	258.0	110	3.08	3.215	19.44	1	0	3	1
## Merc 240D	24.4	4	146.7	62	3.69	3.190	20.00	1	0	4	2
## Merc 230	22.8	4	140.8	95	3.92	3.150	22.90	1	0	4	2
## Fiat 128	32.4	4	78.7	66	4.08	2.200	19.47	1	1	4	1
## Honda Civic	30.4	4	75.7	52	4.93	1.615	18.52	1	1	4	2
## Toyota Corolla	33.9	4	71.1	65	4.22	1.835	19.90	1	1	4	1
## Toyota Corona	21.5	4	120.1	97	3.70	2.465	20.01	1	0	3	1
## Fiat X1-9	27.3	4	79.0	66	4.08	1.935	18.90	1	1	4	1
## Porsche 914-2	26.0	4	120.3	91	4.43	2.140	16.70	0	1	5	2
## Lotus Europa	30.4	4	95.1	113	3.77	1.513	16.90	1	1	5	2
## Volvo 142E	21.4	4	121.0	109	4.11	2.780	18.60	1	1	4	2

```
subset(df, mpg > 20) #me quedo con las filas en las que se verifique que los valores de la columna "mpg" s
```

##	mpg	cyl	disp	hp	drat	wt	qsec	vs	am	gear	carb
## Mazda RX4	21.0	6	160.0	110	3.90	2.620	16.46	0	1	4	4
## Mazda RX4 Wag	21.0	6	160.0	110	3.90	2.875	17.02	0	1	4	4
## Datsun 710	22.8	4	108.0	93	3.85	2.320	18.61	1	1	4	1
## Hornet 4 Drive	21.4	6	258.0	110	3.08	3.215	19.44	1	0	3	1
## Merc 240D	24.4	4	146.7	62	3.69	3.190	20.00	1	0	4	2
## Merc 230	22.8	4	140.8	95	3.92	3.150	22.90	1	0	4	2
## Fiat 128	32.4	4	78.7	66	4.08	2.200	19.47	1	1	4	1
## Honda Civic	30.4	4	75.7	52	4.93	1.615	18.52	1	1	4	2
## Toyota Corolla	33.9	4	71.1	65	4.22	1.835	19.90	1	1	4	1
## Toyota Corona	21.5	4	120.1	97	3.70	2.465	20.01	1	0	3	1
## Fiat X1-9	27.3	4	79.0	66	4.08	1.935	18.90	1	1	4	1
## Porsche 914-2	26.0	4	120.3	91	4.43	2.140	16.70	0	1	5	2
## Lotus Europa	30.4	4	95.1	113	3.77	1.513	16.90	1	1	5	2
## Volvo 142E	21.4	4	121.0	109	4.11	2.780	18.60	1	1	4	2

# Edito Data Frame

## Combinar data frames

Sintaxis:

```
merge(df_1, df_2)
```

## Ejemplo

```
#genero dataframe para agregar al data frame mtcars  
df_personas <- data.frame("Personas" = c(rep(4, 4), rep(2, 16), rep(5, 12))) #genero dataframe de 32 filas  
  
# incorporo nombres al dataframe df_personas igual al nombre de las filas de mtcars  
  
df_personas$Model <- rownames(mtcars)  
mtcars$Model <- rownames(mtcars)
```

```
#combino los data frames
```

```
df_combinada <- merge(mtcars, df_personas)
```

```
head(df_combinada)
```

```
##           Model mpg cyl disp  hp drat   wt  qsec vs am gear carb
## 1      AMC Javelin 15.2   8  304 150 3.15 3.435 17.30  0  0    3    2
## 2 Cadillac Fleetwood 10.4   8  472 205 2.93 5.250 17.98  0  0    3    4
## 3         Camaro Z28 13.3   8  350 245 3.73 3.840 15.41  0  0    3    4
## 4  Chrysler Imperial 14.7   8  440 230 3.23 5.345 17.42  0  0    3    4
## 5         Datsun 710 22.8   4  108  93 3.85 2.320 18.61  1  1    4    1
## 6   Dodge Challenger 15.5   8  318 150 2.76 3.520 16.87  0  0    3    2
## Personas
## 1      5
## 2      2
## 3      5
## 4      2
## 5      4
## 6      5
```

## Borrar una columna

Sintaxis:

```
df$nombre_columna <- NULL
```

## Ejemplo

```
df_combinada$Personas <- NULL  
head(df_combinada)
```

##	Model	mpg	cyl	disp	hp	drat	wt	qsec	vs	am	gear	carb
## 1	AMC Javelin	15.2	8	304	150	3.15	3.435	17.30	0	0	3	2
## 2	Cadillac Fleetwood	10.4	8	472	205	2.93	5.250	17.98	0	0	3	4
## 3	Camaro Z28	13.3	8	350	245	3.73	3.840	15.41	0	0	3	4
## 4	Chrysler Imperial	14.7	8	440	230	3.23	5.345	17.42	0	0	3	4
## 5	Datsun 710	22.8	4	108	93	3.85	2.320	18.61	1	1	4	1
## 6	Dodge Challenger	15.5	8	318	150	2.76	3.520	16.87	0	0	3	2





## Ejercicio

Generar un data frame con dos columnas: una con los valores de la base de 4 parcelas rectangulares y otra con los valores de la altura de esas parcelas.

Calcular la superficie de cada parcela e incorporarla al dataframe como una nueva columna.