

# Data Frames

Rodrigo Negrete Pérez

January 26, 2022

- 1 Data Frames
- 2 Repaso: Creación de DF
- 3 Unidad de observación y variables
- 4 Semillas
- 5 Visualizaciones del DF
- 6 Algunas funciones importantes para DF
- 7 Ejercicios

# Section 1

## Data Frames

# Data Frames

En la sesión anterior vimos vectores. El siguiente paso natural es analizar bases de datos

- No son otra cosa que un conjunto de vectores del mismo tamaño
- Los data frames son un tipo de objeto: algunas funciones requieren que las tablas sean Data frames.

## Section 2

### Repaso: Creación de DF

Aprovechemos lo aprendido para crear un DF. Creemos una base de datos de  $n=4000$  alumnos del ITAM que contenga:

- un numero natural que lo identifique
- sexo
- edad
- carrera: eco, cpol, ri, derecho, conta, mat
- promedio general

Para crear la base de datos necesitamos cada uno de los vectores

Una práctica común es poner el tamaño de la base de datos como una variable, para poder modificarla fácilmente posteriormente. Los primeros dos vectores sabemos cómo hacerlos

```
n<-4000  
id<-1:n  
sex<-sample(c('h','m','o'), n, replace = T)
```

Para crear edades aleatorias podemos usar las funciones de distribución incorporadas en R.

- R tiene incorporadas funciones para generar vectores que provengan de las distribuciones de probabilidad más comunes:
- `runif()` para la distribución uniform
- `rnorm()`
- `rbinom()`
- etc.

Solo se deben especificar los parámetros pertinentes y el tamaño del vector

- Adicionalmente, `qnorm()` se usaría para ver los cuantiles



En nuestro caso conviene usar la uniforme:

```
edad<-runif(n, min = 17, max = 27)
```

Que salgan decimales es extraño, apliquémosle la función piso.

```
edad<-floor(edad)
```

Creemos el resto de los vectores

```
carrera<-sample(c('eco','cpol','ri','derecho','conta','mat'),  
prom<-runif(n, 6, 10)
```

# Creación de Data Frames

Crear data frames es muy sencillo, se hace con la función `data.frame()`

```
df<-data.frame(id, sex, edad, carrera, prom)
```

## Section 3

### Unidad de observación y variables

El recién creado Data Frame aparece en el environment.

The screenshot shows the RStudio Environment pane. At the top, there are tabs for 'Environment', 'History', 'Connections', and 'Tutorial'. Below these is a toolbar with icons for 'Import Dataset', a memory usage indicator showing '252 MiB', and a 'List' button. The 'Environment' tab is active, showing the 'Global Environment'. A search bar is present. The 'Data' section lists the object 'df' as '4000 obs. of 5 variables'. Below this, the 'Values' section displays the structure of the data frame:

Variable	Class	Range	Sample Values
carrera	chr	[1:4000]	"cpol" "mat" "dere..."
edad	num	[1:4000]	23 19 17 24 20 21 ...
id	int	[1:4000]	1 2 3 4 5 6 7 8 9 ...
n			4000
prom	num	[1:4000]	8.9 9.48 6.76 7.67...
sex	chr	[1:4000]	"h" "m" "h" "m" "h..."

# Unidad de observación

- Definición: es la unidad mínima en la que puede cambiar el valor de una variable
- Es cada una de las filas de la base de datos. En nuestro caso, los alumnos.
- Si una categoría aparece varias veces, por sí sola no puede ser la unidad de observación.
- Ejemplos:
  - Diputados
  - País-año
  - alumnos-semester

# Tipos de datos

4 tipos de datos:

- Cross-sectional/ de corte transversal
- Series de tiempo
- pooled cross sections
- panel/ “time series cross-sectional”

# Cross\_sectional/ de corte transversal

- Cada unidad aparece una sola vez. Es una fotografía de la unidad tomada en un punto particular del tiempo
- NO HAY UNIDADES DE TIEMPO
- Ejemplos:
  - Encuestas



# Series de tiempo

- Observaciones de UNA entidad a lo largo del tiempo.
- El orden importa
- Ejemplos:
  - PIB
  - Tasas de interés

# Pooled cross section

- Combinaciones de al menos dos *cross-sections*
- mismas variables son analizadas para al menos dos periodos de tiempo, pero sin seguir a las mismas unidades.
- Seguir individuos es costoso: mejor tomas muestras representativas en distintos periodos de tiempo.
- Ejemplos:
  - LAPOP
  - ENIGH

# Panel / Time series cross-sectional

- Una serie de tiempo para cada miembro cross-sectional
- un conjunto de entidades es observado varias veces en el tiempo
- Ejemplos:
  - Líderes mundiales
  - Datos OCDE

# Observaciones y variables

- Las observaciones son las filas: las entidades mínimas que estamos observando
- Las **variables** son aquello que estamos observando de la unidad de observación: las columnas

# Tipos de datos: resumen

(a) *cross-sectional*

id	time	$x_1$	$x_2$	$x_3$	...	$x_n$
A	1				...	
B	1				...	
C	1				...	
D	1				...	
E	1				...	
F	1				...	
G	1				...	
H	1				...	
...	...	...	...	...	...	...
N	1				...	

(b) *time series*

id	time	$x_1$	$x_2$	$x_3$	...	$x_n$
A	1				...	
A	2				...	
A	3				...	
A	4				...	
A	5				...	
A	6				...	
A	7				...	
A	8				...	
...	...	...	...	...	...	...
A	T				...	

(c) *pooled cross-sectional*

id	time	$x_1$	$x_2$	$x_3$	...	$x_n$
A	1				...	
B	1				...	
C	1				...	
D	2				...	
E	2				...	
F	2				...	
...	...	...	...	...	...	...
X	T				...	
Y	T				...	
N	T				...	

(d) *panel*

id	time	$x_1$	$x_2$	$x_3$	...	$x_n$
A	1				...	
A	...				...	
A	T				...	
B	1				...	
B	...				...	
B	T				...	
...	...	...	...	...	...	...
N	1				...	
N	...				...	
N	T				...	

## Section 4

### Semillas

Podemos hacer click sobre el df en el environment para que nos lo muestre R

The screenshot shows the RStudio IDE. On the left, a data frame 'df' is displayed in a table view with columns: id, sex, edad, carrera, and prom. The table contains 12 rows of data. On the right, the Environment pane shows the variable 'df' with 4000 observations and 5 variables. Below this, the 'Values' section shows the data types and ranges for each variable: 'carrera' is character [1:4000], 'edad' is numeric [1:4000], 'id' is integer [1:4000], 'n' is numeric [1:4000], 'prom' is numeric [1:4000], and 'sex' is character [1:4000].

	id	sex	edad	carrera	prom
1	1	m	25	ri	9.322858
2	2	h	20	conta	7.975751
3	3	h	22	ri	9.769267
4	4	h	18	ri	9.445680
5	5	h	20	conta	7.548371
6	6	m	26	eco	7.572883
7	7	m	22	mat	7.369928
8	8	m	21	ri	8.535115
9	9	m	19	eco	8.503445
10	10	m	22	conta	7.246320
11	11	h	24	derecho	7.784699
12	12	h	24	cpol	9.197110

- ¡Pero cada quien obtendría un df distinto!
- Las variables de sexo, edad, carrera y promedio las obtuvimos usando cierta aleatoriedad



# Semillas

- En realidad, R no utiliza valores aleatorios, sino pseudo-aleatorios a través de algoritmos.
- Gracias a dichos algoritmos podemos replicar los valores pseudo-aleatorios usando la función `set.seed()`

```
set.seed(2020)
```

si corremos la función `set.seed()` y el mismo número, y luego corremos la misma base de datos, deberíamos obtener el mismo `df`.

```
df<-data.frame(  
  id=1:n,  
  sex=sample(c('h','m','o'), n, replace = T),  
  edad=floor(runif(n, min=17, max=27)),  
  carrera=sample(c('eco','cpol','ri','derecho','conta','mat'),  
  prom=runif(n, 6, 10)  
)
```

- Nota que también podemos crear el `df` directamente, sin generar las variables.

# Hacia una mayor replicabilidad

Siempre que trabajemos con datos aleatorios en un proyecto hay que incluir la semilla para que nuestros resultados sean replicables.

## Section 5

# Visualizaciones del DF

- Como vimos, podemos hacer click en el df en el environment para ver el df.
- Podemos nombrarlo en la consola
- Podemos usar funciones preestablecidas para darnos una mejor idea de los datos que estamos viendo.

# Funciones para visualizar

- `summary()` Muestra un pequeño resumen para cada variable: media, max, min.
- `head()` Muestra las primeras observaciones
- `str()` muestra algunas variables y qué tipo de objeto son.

# Particiones de Df

- Podemos extraer los vectores de las variables con '\$'

```
df$prom
```

```
##      [1] 7.571058 9.999865 6.506111 7.422582 6.238837 9.21541
##      [9] 7.729670 7.318772 9.305000 6.674212 6.994838 8.19720
##     [17] 9.817460 6.915154 9.769244 7.510031 9.900142 8.18820
##     [25] 8.536120 8.917404 6.027189 9.179646 8.412015 7.13240
##     [33] 6.555146 7.278672 7.066735 8.444566 7.910612 7.84509
##     [41] 8.757885 8.698615 6.740078 8.706213 8.099774 7.66212
##     [49] 9.851144 6.862523 6.132346 6.009321 7.498169 8.35742
##     [57] 9.938698 6.427188 8.061951 6.174165 8.634576 7.38475
##     [65] 6.219950 7.232165 9.782402 9.889615 6.089259 9.13992
##     [73] 8.913660 9.205672 9.330211 6.972007 6.220936 8.91858
##     [81] 9.024586 8.630223 7.179790 8.763815 6.312621 6.96495
##     [89] 8.003474 9.957010 6.579256 6.655421 9.574871 7.63426
##     [97] 8.716999 8.789999 7.510515 7.887007 7.445487 8.81875
```

Podemos operar con este vector como antes

```
df$prom[5]
```

```
## [1] 6.238837
```

```
mean(df$prom)
```

```
## [1] 7.993034
```



# Slices de DF

Podemos especificar partes del df

```
df[filas, columnas]
```

- Por ejemplo, si queremos las primeras dos filas y las columnas 2-3 y 5

```
df[1:2, c(2:3,5)]
```

```
##      sex edad      prom
## 1     o   25  7.571058
## 2     m   18  9.999865
```

## Section 6

### Algunas funciones importantes para DF

## with()

A menudo, conviene usar la función `with()` para operar con los vectores en lugar de llamarlos con `$`

- Por ejemplo, calculemos el promedio de los promedios.

```
with(df, mean(prom))
```

```
## [1] 7.993034
```

## subset()

`subset()` permite quedarnos con observaciones que cumplan ciertas características

- Por ejemplo, creemos un df solo con las observaciones de mujeres que estudian RRII

```
df_female_ri<-subset(df, sex=='m' & carrera=='ri')
```

- recuerden sus operadores lógicos
- Reemplacen esta función con `filter(df, condición)` cuando veamos `dplyr`

## ifelse()

Las variables dicotómicas (dummies) son fundamentales para la econometría. Podemos crearlas fácilmente con la función `ifelse()`

`ifelse(condición a cumplir, qué pone R en caso de cumplirla, qué pone en caso de que no)`

- Podemos crear variables con `df$new.var<-`
- `ifelse()` también funciona para vectores independientes

Creemos una dummy que indique si es una alumna

```
df$female<-ifelse(sex=='m', 1, 0)
```

- *Crea una columna llamada female: pon un uno en caso de ser mujer, pon un 0 e.o.c*

## Section 7

### Ejercicios

## Con el df creado

- Calcula el promedio de los hombres que estudian economía.
- Calcula la proporción de mujeres que estudian matemáticas. Para esto, recuerda que el promedio de una dummy es la proporción de observaciones que cumplen la característica.
- ¿Cuál es el id del hombre con promedio más alto en la carrera de Ciencia Política? Para esto, recuerda la función `max()` o `sort()`
- Crea una variable que identifique con un número la carrera: 1=eco, 2=ri, 3=e.o.c. Para esto, concatena `ifelse()`
- Obtén el DF de las mujeres que estudian RRII, pero con slices.



## Creación de DF tipo panel

Creemos una base de datos de vacunación. Hay tres tipos de individuos: niños, adultos y adultos mayores. Hay tres periodos de tiempo: 1,2,3. Los adultos mayores se vacunan en el primer periodo; los adultos en el segundo; niños en el tercero. Crea una base de datos tipo panel con un identificador, una variable de edad (con distribución uniforme y redondeada al entero menor), una variable de tiempo y una dummy que valga uno si al individuo-tiempo le corresponde una vacuna.