#### **Data Frames**

Rodrigo Negrete Pérez

January 26, 2022

- Data Frames
- Repaso: Creación de DF
- Unidad de observación y variables
- 4 Semillas
- Visualizaciones del DF
- 6 Algunas funciones importantes para DF
- 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

5 / 41

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

- un numero natural que l@ 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)</pre>
```

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.

#### Creemos el resto de los vectores

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

#### Creación de Data Frames

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

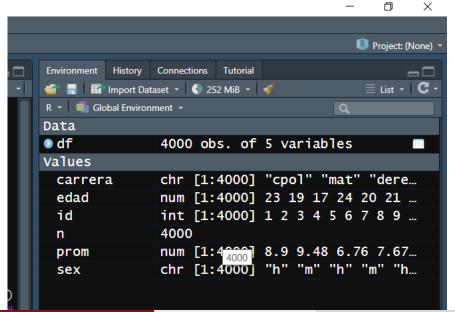
df<-data.frame(id, sex, edad, carrera, prom)</pre>

### Section 3

Unidad de observación y variables

12 / 41

#### El recién creado Data Frame aparece en el environment.



#### 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-semestre

# 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

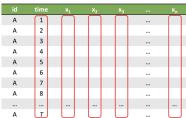
20 / 41

## Tipos de datos: resumen



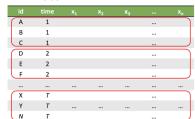
id	time	$\mathbf{x_1}$	x <sub>2</sub>	x <sub>3</sub>	 $\mathbf{x}_n$
Α	1				
В	1				
С	1				
D	1				
E	1				
F	1				
G	1				
Н	1				
N	1				

 $x_3$  ...  $x_n$  id time  $x_1$ 



(b) time series

(c) pooled cross-sectional



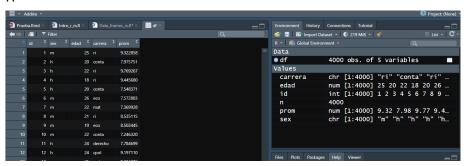
(d) panel

(4) parior									
id	time	X <sub>1</sub>	X <sub>2</sub>	X <sub>3</sub>		× <sub>n</sub>			
A	1								
Α									
A	Т								
В	1					$\overline{}$			
В									
В	Т								
N	1								
N									
N	Т								

## Section 4

# Semillas

Podemos hacer click sobre el df en el environment para que nos lo muestre  ${\sf R}$ 



- ¡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)
  )</pre>
```

 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.

27 / 41

### Section 5

## Visualizaciones del DF

28 / 41

- 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
```

##

##

##

##

##

##

##

```
## [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 7.38478
```

[1] 7.571058 9.999865 6.506111 7.422582 6.238837 9.2154: [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

[65] 6.219950 7.232165 9.782402 9.889615 6.089259 9.1399

8.913660 9.205672 9.330211 6.972007 6.220936 8.91858 9.024586 8.630223 7.179790 8.763815 6.312621 6.96498

8.003474 9.957010 6.579256 6.655421 9.574871 7.63426

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

34 / 41

# 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')</pre>
```

- 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<-</li>
- ifelse() también funciona para vectores independietes

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 audltos 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 distribucion uniforme y redondeada al entero menor), una variable de tiempo y una dummy que valga uno si al individio-tiempo le corresponde una vacuna.