

Lectura 1

Inferencia de modelos estadísticos

Definición

-> La Real Academia Española define **inferencia** como “**acción y efecto de inferir**”

-> La **estadística inferencial** es una rama de la estadística que busca obtener una conclusión para un conjunto de individuos o elementos (denominado **población**) a partir de información recolectada de un subconjunto de éste (llamado **muestra**).

Variables, parámetros y estadísticos

Numéricas: Pueden tomar muchos valores numéricos

-> **Continuas:** pueden tomar cualquier valor (en un intervalo) del conjunto de los reales. (ej: estatura)

-> **Discretas:** No es posible que tomen cualquier valor (ej: valores enteros no negativos)

Categóricas: Sólo pueden tomar un valor de entre un conjunto acotado. Cada posible valor se denomina nivel.

-> **Nominales:** no existe un orden natural entre los niveles.

-> **Ordinales:** existe un orden natural entre los niveles.

Tener diferentes variables significa que debemos medirlas con distintas clases de escalas, las cuales se distinguen por sus propiedades y los tipos de operaciones que permiten.

-> **Escala nominal**

-> **Escala ordinal o de rangos**



-> **Escala de intervalo**

-> **Escala de razón**

Las variables pueden ser:

- 1) **Independientes:** No existe asociación o relación entre las variables
- 2) **Dependientes:** Existe una relación o asociación entre las variables. Puede existir:

-> **Asociación positiva:** si una variable crece, la otra también lo hace.

-> **Asociación negativa:** si una variable crece, la otra decrece.



Conociendo R

Importación de datos

id	género	estatura	escalafón	servicio	antigüedad	rama
1	M	1.77	S	89.91	15	E
2	M	1.97	O	65.14	30	C
3	F	1.65	O	97.03	12	A
4	M	1.82	S	76.29	9	A
5	F	1.73	S	69.46	7	M
6	M	1.78	S	97.67	21	E
7	M	1.87	O	72.09	27	C
8	F	1.91	S	94.53	11	A

(a) Texto plano delimitado por tabulaciones.

```
id,género,estatura,escalafón,servicio,antigüedad,rama
1,M,1.77,S,89.91,15,E
2,M,1.97,O,65.14,30,C
3,F,1.65,O,97.03,12,A
4,M,1.82,S,76.29,9,A
5,F,1.73,S,69.46,7,M
6,M,1.78,S,97.67,21,E
7,M,1.87,O,72.09,27,C
8,F,1.91,S,94.53,11,A
```

(b) Valores separados por comas (inglés).

```
id;género;estatura;escalafón;servicio;antigüedad;rama
1;M;1.77;S;89.91;15;E
2;M;1.97;O;65.14;30;C
3;F;1.65;O;97.03;12;A
4;M;1.82;S;76.29;9;A
5;F;1.73;S;69.46;7;M
6;M;1.78;S;97.67;21;E
7;M;1.87;O;72.09;27;C
8;F;1.91;S;94.53;11;A
```

(c) Valores separados por punto y comas (español).

Figura 1.2: formatos de archivo para importar datos en R.

->

Script 1.1: sentencias para importar un conjunto de datos.

```
1 # Cargar un conjunto de datos disponible en R.
2 datos1 <- mtcars
3
4 # Importar desde un archivo de texto plano delimitado por tabuladores.
5 datos2 <- read.delim(file.choose())
6
7 # Importar desde un archivo de valores separados por coma
8 # en formato inglés (figura 1.2 b).
9 datos3 <- read.csv("C:\\Inferencia\\ejemplo1-csv-eng.csv")
10
11 # Configurar carpeta de trabajo
12 setwd("C:\\Inferencia")
13
14 # Importar desde un archivo de valores separados por coma
15 # en formato español (figura 1.2 c).
16 datos4 <- read.csv2("ejemplo1-csv-esp.csv")
17
18 # Mostrar las primeras 6 filas del conjunto de datos
19 # almacenado en datos1.
20 head(datos1)
21
22 # Mostrar las últimas 6 filas del conjunto de datos
23 # almacenado en datos1.
24 tail(datos1)
```

Construcción matriz de datos

as.Date() -> convertir formato fecha

data.frame() -> matriz de datos

Si queremos guardar nuestra matriz de datos en un archivo separado por comas en formato inglés, podemos hacerlo mediante la función write.csv(), que funciona del mismo modo que write.csv2() .

Script 1.3: construir un dataframe.

```
1 # Crear un vector de strings y guardarlo en la variable nombre.
2 nombre <- c("Alan Brito Delgado",
3             "Zacarías Labarca del Río",
4             "Elsa Payo Maduro")
5
6 # Crear un vector de fechas y guardarlo en la variable
7 # fecha_nacimiento.
8 fecha_nacimiento <- as.Date(c("2008-1-25", "2006-10-4", "2008-3-27"))
9
10 # Crear tres vectores de reales entre 1.0 y 7.0 y guardarlos
11 # en prueba_i, respectivamente.
12 prueba_1 <- c(5.5, 3.4, 4.5)
13 prueba_2 <- c(3.2, 4.7, 4.1)
14 prueba_3 <- c(4.8, 4.3, 5.1)
15
16 # Construir un data frame a partir de los vectores anteriores y
17 # guardarlo en la variable dataframe.
18 dataframe <- data.frame(nombre,
19                          fecha_nacimiento,
20                          prueba_1,
21                          prueba_2,
22                          prueba_3,
23                          stringsAsFactors = FALSE)
24
25 # Guardar un dataframe en un archivo csv (formato español).
26 write.csv2(dataframe, "C:/Inferencia/Ejemplo.csv", row.names = FALSE)
```

Modificación de una matriz de datos

Muchas veces tendremos la necesidad de modificar la matriz de datos. Algunas tareas, como agregar o quitar una columna o un observación pueden hacerse de manera bastante sencilla, como ilustra el script 1.4.

Script 1.4: modificaciones sencillas de una matriz de datos.

```
1 # Leer un dataframe desde archivo csv.
2 datos <- read.csv2("C:/Inferencia/Ejemplo.csv", stringsAsFactors = FALSE)
3
4 # Eliminar del data frame la columna fecha_nacimiento.
5 dataframe$fecha_nacimiento <- NULL
6
7 # Agregar al data frame la columna edad.
8 dataframe$edad <- c(23, 25, 23)
9
10 # Crear una nueva observación.
11 nueva <- data.frame(nombre="Elba Calao del Río",
12                     prueba_1 = 6.4,
13                     prueba_2 = 2.3,
14                     prueba_3 = 4.6,
15                     edad = 24)
16
17 # Agregar la nueva observación al data frame.
18 dataframe <- rbind(dataframe, nueva)
19
20 # Eliminar las primeras 3 observaciones del data frame.
21 dataframe <- dataframe[-c(1:3),]
22
23 # Guardar el dataframe en un archivo csv .
24 write.csv2(dataframe, "C:/Inferencia/Ejemplo_mod.csv", row.names = FALSE)
```

Paquete dplyr

Script 1.5: modificación de una matriz de datos con el paquete dplyr.

```
1 library(dplyr)
2
3 # Cargar dataframe iris incluido en R.
4 datos <- iris
5
6 # Seleccionar observaciones correspondientes a la especie versicolor.
7 versicolor <- datos %>% filter(Species == "versicolor")
8
9 # Seleccionar observaciones de la especie versicolor cuyos sépalos tengan una
10 # longitud igual o superior a 6 cm.
11 largas <- datos %>% filter(Species == "versicolor" & Sepal.Length >= 6)
12
13 # Seleccionar la especie y variables relativas a los pétalos.
14 petalos <- datos %>% select(Species, starts_with("Petal"))
15
16 # Seleccionar variables de ancho y la especie.
17 anchos <- datos %>% select(ends_with("Width"), Species)
18
19 # Agregar al conjunto de datos de los pétalos una nueva variable con la razón
20 # entre el largo y el ancho de éstos.
21 petalos <- petalos %>% mutate(Species, Petal.Width,
22                               Petal.Ratio = Petal.Length / Petal.Width)
23
24 # Ordenar el conjunto de datos de pétalos en forma descendente según la razón
25 # de los pétalos.
26 petalos <- petalos %>% arrange(desc(Petal.Ratio))
27
28 # Ordenar el conjunto de datos de pétalos en forma ascendente según el largo de
```

Q

Paquete tidyr

Script 1.6: modificación de una matriz de datos con el paquete tidyr.

```
1 library(dplyr)
2 library(tidyr)
3
4 # Crear el data frame.
5 Instancia <- 1:6
6 Quicksort <- c(23.2, 22.6, 23.4, 23.3, 21.8, 23.9)
7 Bubblesort <- c(31.6, 29.3, 30.7, 30.8, 29.8, 30.3)
8 Radixsort <- c(30.1, 28.4, 28.7, 28.3, 29.9, 29.1)
9 Mergesort <- c(25.0, 25.7, 25.7, 23.7, 25.5, 24.7)
10 datos <- data.frame(Instancia, Quicksort, Bubblesort, Radixsort, Mergesort)
11
12 # Mostrar las primeras filas de la matriz de datos.
13 cat("Datos originales\n")
14 print(head(datos))
15 cat("\n")
16
17 # Convertir la matriz de datos a formato largo.
18 datos_largos <- datos %>% pivot_longer(c("Quicksort", "Bubblesort",
19                                         "Radixsort", "Mergesort"),
20                                         names_to = "Algoritmo",
21                                         values_to = "Tiempo")
22
23 # Mostrar las primeras filas de la matriz de datos largos.
24 cat("Datos largos\n")
25 print(head(datos_largos))
26 cat("\n")
27
28 # Convertir la matriz de datos largos a formato ancho.
29 datos anchos <- datos_largos %>% pivot_wider(names_from = "Algoritmo",
30                                              values_from = "Tiempo")
31
32 # Mostrar las primeras filas de la matriz de datos anchos.
33 cat("Datos anchos\n")
34 print(head(datos anchos))
35 cat("\n")
```

Script 1.7: modificación del conjunto de datos mtcars para facilitar su comprensión.

```
1 library(dplyr)
2
3 # Cargar conjunto de datos.
4 datos <- mtcars
5
6 # Renombrar columnas.
7 datos <- datos %>% rename(Rendimiento = mpg, Cilindrada = cyl,
8                           Desplazamiento = disp, Potencia = hp,
9                           Eje = drat, Peso = wt, Cuarto_milla = qsec,
10                          Motor = vs, Transmision = am, Cambios = gear,
11                          Carburadores = carb)
12
13 # Dar formato categórico a las variables Motor y Transmision, renombrando
14 # sus niveles.
15 datos[["Motor"]] <- factor(datos[["Motor"]], levels = c(0, 1),
16                            labels = c("V", "Recto"))
17
18 datos[["Transmision"]] <- factor(datos[["Transmision"]], levels = c(0, 1),
19                                  labels = c("Automático", "Manual"))
20
21 # Dar formato ordinal a las variables Cilindrada y Cambios, renombrando
22 # sus niveles.
23 datos[["Cilindrada"]] <- factor(datos[["Cilindrada"]], levels = c(4, 6, 8),
24                                  labels = c("4 cilindros", "6 cilindros",
25                                                "8 cilindros"),
26                                  ordered = TRUE)
27
28 datos[["Cambios"]] <- factor(datos[["Cambios"]], levels = c(3, 4, 5),
29                               labels = c("3 cambios", "4 cambios", "5 cambios"),
30                               ordered = TRUE)
31
32 write.csv2(datos, "C:/Inferencia/Mtcars.csv")
```