**DATAFRAMES**

**Introducción**

Por mucho, la estructura de datos R más importante utilizada en el aprendizaje automático es el marco de datos o dataframes.

Es una estructura análoga a una hoja de cálculo o una base de datos, ya que tiene filas y columnas de datos.

En términos de R, un marco de datos puede entenderse como una lista de vectores o factores, cada uno con exactamente el mismo número de valores.

Debido a que el marco de datos es literalmente una lista de objetos de tipo vectorial, combina aspectos de ambos vectores y listas.

**Recapitulando**

Ingresemos nuevamente los datos con los que hemos venido trabajando.

subject\_name **<-** c**(**"John Doe", "Jane Doe", "Steve Graves"**)**

temperature **<-** c**(**98.1, 98.6, 101.4**)**

flu\_status **<-** c**(FALSE**, **FALSE**, **TRUE)**

gender **<-** factor**(**c**(**"MALE", "FEMALE", "MALE"**))**

blood **<-** factor**(**c**(**"O", "AB", "A"**)**, levels **=** c**(**"A", "B", "AB", "O"**))**

symptoms **<-** factor**(**c**(**"SEVERE", "MILD", "MODERATE"**)**,levels **=** c**(**"MILD", "MODERATE", "SEVERE"**)**, ordered **=** **TRUE)**

**Creando un DataFrame**

Vamos a crear un marco de datos para nuestro conjunto de datos de pacientes.

Usando los vectores de datos del paciente que creamos anteriormente, la función data.frame() los combina en un marco de datos así:

pt\_data **<-** data.frame**(**subject\_name, temperature, flu\_status, gender, blood, symptoms, stringsAsFactors **=** **FALSE)**

Es posible que note algo nuevo en el código anterior. Incluimos un parámetro adicional: stringsAsFactors = FALSE. Si no especificamos esta opción, R convertirá automáticamente cada vector de caracteres en un factor.

Esta característica es ocasionalmente útil, pero a veces también es injustificada.

Aquí, por ejemplo, el campo subject\_name definitivamente no son datos categóricos, ya que los nombres no son categorías de valores.

Por lo tanto, establecer la opción stringsAsFactors en FALSE nos permite convertir vectores de caracteres a factores solo donde tiene sentido para el proyecto.

**Estructura**

Cuando mostramos el marco de datos pt\_data, vemos que la estructura es bastante diferente de las estructuras de datos con las que trabajamos anteriormente:

**>** pt\_data

subject\_name temperature flu\_status gender blood symptoms

1 John Doe 98.1 **FALSE** MALE O SEVERE

2 Jane Doe 98.6 **FALSE** FEMALE AB MILD

3 Steve Graves 101.4 **TRUE** MALE A MODERATE

**Comparando**

En comparación con los vectores, factores y listas unidimensionales, un marco de datos tiene dos dimensiones y se muestra en formato de matriz.

Este marco de datos en particular tiene una columna para cada vector de datos del paciente y una fila para cada paciente.

En términos de aprendizaje automático, las columnas del marco de datos son las características o atributos y las filas son los ejemplos, muestras o como también se le llama observaciones (En especial a mi persona le van a escuchar bastante llamarlas de ese modo por pura costumbre, ustedes las pueden llamar ejemplos o muestras).

**Extrayendo Información**

Para extraer columnas enteras (vectores) de datos, podemos aprovechar el hecho de que un marco de datos es simplemente una lista de vectores.

Similar a las listas, la forma más directa de extraer un solo elemento es refiriéndose a él por su nombre. Por ejemplo, para obtener el vector subject\_name, escriba:

**>** pt\_data**$**subject\_name

**[**1**]** "John Doe" "Jane Doe" "Steve Graves"

También similar a las listas, se puede usar un vector de nombres para extraer varias columnas de un marco de datos:

**>** pt\_data**[**c**(**"temperature", "flu\_status"**)]**

temperature flu\_status

1 98.1 **FALSE**

2 98.6 **FALSE**

3 101.4 **TRUE**

**Alternativas**

Cuando accedemos al marco de datos de esta manera, el resultado es un marco de datos que contiene todas las filas de datos para todas las columnas solicitadas.

Alternativamente, el comando pt\_data[2: 3] también extraerá las columnas de temperatura y flu\_status.

temperature flu\_status

1 98.1 **FALSE**

2 98.6 **FALSE**

3 101.4 **TRUE**

Sin embargo, solicitar las columnas por nombre da como resultado un código R claro y fácil de mantener que no se romperá si el marco de datos se reestructura en el futuro.

**Excepciones**

Para extraer valores en el marco de datos, se utilizan métodos como los [] para acceder a valores en vectores. Sin embargo, hay una excepción importante.

Debido a que el marco de datos es **bidimensional**, se deben especificar tanto las filas como las columnas que se desea extraer.

Las filas se especifican primero, seguidas de una coma y luego las columnas en un formato como este: **[filas, columnas]**.

Al igual que con los vectores, las filas y columnas se cuentan comenzando en uno.

**Ejemplo**

Para extraer el valor en la primera fila y la segunda columna del marco de datos del paciente (el valor de temperatura para John Doe), use el siguiente comando:

**>** pt\_data**[**1, 2**]**

**[**1**]** 98.1

**Mas datos**

Si desea más de una fila o columna de datos, especifique vectores para las filas y columnas deseadas. El siguiente comando extraerá los datos de las filas primera y tercera y las columnas segunda y cuarta:

**>** pt\_data**[**, 1**]**

**[**1**]** "John Doe" "Jane Doe" "Steve Graves"

Para extraer todas las columnas de la primera fila, use el siguiente comando:

**>** pt\_data**[**1, **]**

subject\_name temperature flu\_status gender blood symptoms

1 John Doe 98.1 **FALSE** MALE O SEVERE

Para extraer todo, use el siguiente comando:

**>** pt\_data**[**,**]**

subject\_name temperature flu\_status gender blood symptoms

1 John Doe 98.1 **FALSE** MALE O SEVERE

2 Jane Doe 98.6 **FALSE** FEMALE AB MILD

3 Steve Graves 101.4 **TRUE** MALE A MODERATE

**Otros Métodos**

También se pueden utilizar otros métodos para acceder a valores en listas y vectores para recuperar filas y columnas de marcos de datos. Por ejemplo, se puede acceder a las columnas por nombre en lugar de por posición, y los signos negativos se pueden usar para excluir filas o columnas de datos.

Como ejemplo les pongo el siguiente código:

**>** pt\_data**[**c**(**1, 3**)**, c**(**"temperature", "gender"**)]**

temperature gender

1 98.1 MALE

3 101.4 MALE

Que es el equivalente a:

**>** pt\_data**[-**2, c**(-**1, **-**3, **-**5, **-**6**)]**

temperature gender

1 98.1 MALE

3 101.4 MALE

**Como todo en esta vida**

Para lograr familiarizarnos con los dataframes hay que hacer mucha práctica, en especial con operaciones similares como las que les he puesto en los ejemplos, espero que, al tener este documento en sus manos, o frente a sus ojos digitalmente escribiendo, ya tengan acceso a las fuentes de datos de la institución y puedan comenzar a interactuar con los datos propios.

Estos tipos de operaciones son cruciales para gran parte del trabajo que haremos en los próximos capítulos.

**PRACTICA**

**Introducción:** En comparación con los vectores, factores y listas unidimensionales, un marco de datos tiene dos dimensiones y se muestra en formato de matriz.

**Ejercicio 1**

Cree el siguiente Data Frame, luego invierta el sexo para todos los individuos.

# Age Height Weight Sex

#Alex 25 177 57 M

#Lilly 31 163 69 M

#Mark 23 190 83 F

#Oliver 52 179 75 F

#Martha 76 163 70 M

#Lucas 49 183 83 F

#Caroline 26 164 53 M

**Ejercicio 2**

Cree este Data Frame (asegúrese de importar la variable Working como carácter y no como factor).

# Working

#Alex Yes

#Lilly No

#Mark No

#Oliver Yes

#Martha Yes

#Lucas No

#Caroline Yes

Agregue este Data Frame en forma de columna al anterior que realizo en el ejercicio 1, utilice la función cbind.

a) ¿Cuántas filas y columnas tiene el nuevo Data Frame?

b) ¿Qué clase de datos hay en cada columna?

**Tip:** Utilice las funciones nrow(), ncol(), dim(), sapply(), str() como usted considere conveniente.

**Ejercicio 3**

Compruebe qué clase de datos es conjunto de datos incorporado state.center y conviértalo a Data Frame.

**TIP:** Cargue desde R studio la clase de la siguiente forma:

class(state.center)

**Ejercicio 4**

Cree un Data Frame simple a partir de 3 vectores con nombre v, b, y n. Cree en Data Frame etiquetándolo así Age = v, Class = b, Grade = n . Ordene todo el Data Frame por la primera columna para que quede igual que el resultado siguiente:

## Age Class Grade

## 6 30 C 57

## 7 31 A 64

## 8 32 B 59

## 9 33 C 73

## 5 41 B 61

## 4 42 A 71

## 3 43 C 70

## 2 44 B 63

## 1 45 A 62

**Tip:**

Utilice LETTERS para “v”, de esta forma “LETTERS[rep()]”

Utilice la funcion round() para eliminar los decimales y rnorm() para crear en n, 9 observaciones, con un promedio de 65 y una desviación estándar de 5.

**Ejercicio 5**

Cree un Data Frame a partir de una matriz, con número del 1 a las 20 y 5 columnas, cambie los nombres de las filas para que cada fila diga id\_i (donde i es el número de la fila) y cambie los nombres de las columnas a variable\_i (donde i es el número de la columna).

Es decir, para la columna 1 dirá variable\_1, y para la fila 2 dirá id\_2 y así sucesivamente.

**Ejercicio 6**

Para este ejercicio, usaremos el conjunto de datos VADeaths (incorporado).

class(VADeaths)

VADeaths

a) Asegúrese de que el objeto sea un Data Frame, si no lo cambia a un Data Frame.

b) Cree una nueva variable, llamada Total, que es la suma de cada fila.

c) Cambie el orden de las columnas para que el total sea la primera variable.

**Ejercicio 7**

Para este ejercicio usaremos el conjunto de datos (incorporado) state.x77.

**TIP:** en este caso R ya posee los datos en memoria, únicamente teclee state.x77 para llamar la información.

a) Asegúrese de que el objeto sea un Data Frame, si no lo cambia a un Data Frame.

b) Averigüe cuántos estados tienen un ingreso inferior a 4300.

c) Averiguar cuál es el estado con mayor ingreso.

**Ejercicio 8**

Con el conjunto de datos swiss, cree un Data Frame de solo las filas 1, 2, 3, 10, 11, 12 y 13, y solo las variables Examination, Education y Infant.Mortality.

class(swiss)

a) La mortalidad infantil de Sarine es incorrecta, debe ser una NA, cambiarla.

b) Cree una fila que será la suma total de la columna, asígnele el nombre Total.

c) Crear una nueva variable que será la proporción del examen (examen / total)

**Ejercicio 9**

Cree un Data Frame con los conjuntos de datos state.abb, state.area, state.division, state.name, state.region. Los nombres de las filas deben ser los nombres de los estados.

a) Cambie el nombre de los nombres de las columnas para que solo aparezcan las primeras 3 letras (por ejemplo, States.abb será abb).

**Ejercicio 10**

state.x77

Agregue el Data Frame anterior en forma de columna a state.x77

a) Eliminar la variable div.

b) También elimine las variables Life Exp, HS Grad, Frost, abb y are.

c) Agregue una variable al Data Frame que debería categorizar el nivel de analfabetismo: [0,1) es bajo, [1,2) es algo, [2, inf) es alto.

d) Averigüe qué estado del oeste, con bajo analfabetismo, tiene el ingreso más alto y cuál es ese ingreso.