

Tarea25

Yimmy Eman

2022-07-17

Pregunta 1

Haz un dibujo de las siguientes listas anidadas:

• list(a, b, list(c, d), list(e, f)) • list(list(list(list(list(list(a))))))

Pregunta 2

¿Qué pasa si haces un subconjunto de una tibble con la sintaxis de cómo si se tratara de una lista? Infiere entonces el comportamiento que permite diferenciar una tibble de una lista.

Subdividir un tibble funciona de la misma manera que una lista; un marco de datos se puede considerar como una lista de columnas. La diferencia clave entre una lista y un tibble es que todos los elementos (columnas) de un tibble deben tener la misma longitud (número de filas). Las listas pueden tener vectores con diferentes longitudes como elementos.

Pregunta 3

¿Qué devuelve la función `hms::hms(3600)` y cómo lo imprime? Investiga sobre qué tipo de dato primitivo está construido este vector aumentado y qué atributos utiliza.

```
x <- hms::hms(3600)
class(x)
```

```
## [1] "hms" "difftime"
```

```
x
```

```
## 01:00:00
```

Pregunta 4

Intenta crear una tibble con columnas que tengan diferente longitud. ¿Qué ocurre?

```
#tibble(x = 1, y = 1:5)
```

```
#tibble(x = 1:3, y = 1:4)
# Error:
#! Tibble columns must have compatible sizes.
# Size 3: Existing data.
# Size 4: Column `y`.
#i Only values of size one are recycled.
#Backtrace:
# 1. tibble::tibble(x = 1:3, y = 1:4)
# 2. tibble::tibble_quos(xs, .rows, .name_repair)
# 3. tibble::vectbl_recycle_rows(res, first_size, j, given_col_names[[j]])
```

Pregunta 5

Según lo que has practicado en esta tarea, ¿está bien tener una lista como columna de una tibble? ¿Por qué?

El mensaje de error se refería a vectores que tenían diferentes longitudes. Pero no hay nada que impida que un tibble tenga vectores de diferentes tipos: dobles, carácter, enteros, lógico, factor, fecha. Los últimos siguen siendo atómicos, pero tienen atributos adicionales. Entonces, tal vez no haya problemas con un vector de lista siempre que tenga la misma longitud.