

# Tarea: Las Tibbles de tidyverse

Jesus Mudarra Luján

2023-01-08

## Pregunta 1

¿Tibble o no tibble? ¿Cómo sabes si un objeto es una tibble o no?

Pista: imprime los objetos `mtcars` por un lado y `nycflights13::flights` que son respectivamente un data frame y una tibble.

```
print(mtcars)
```

	mpg	cyl	disp	hp	drat	wt	qsec	vs	am	gear	carb
Mazda RX4	21.0	6	160.0	110	3.90	2.620	16.46	0	1	4	4
Mazda RX4 Wag	21.0	6	160.0	110	3.90	2.875	17.02	0	1	4	4
Datsun 710	22.8	4	108.0	93	3.85	2.320	18.61	1	1	4	1
Hornet 4 Drive	21.4	6	258.0	110	3.08	3.215	19.44	1	0	3	1
Hornet Sportabout	18.7	8	360.0	175	3.15	3.440	17.02	0	0	3	2
Valiant	18.1	6	225.0	105	2.76	3.460	20.22	1	0	3	1
Duster 360	14.3	8	360.0	245	3.21	3.570	15.84	0	0	3	4
Merc 240D	24.4	4	146.7	62	3.69	3.190	20.00	1	0	4	2
Merc 230	22.8	4	140.8	95	3.92	3.150	22.90	1	0	4	2
Merc 280	19.2	6	167.6	123	3.92	3.440	18.30	1	0	4	4
Merc 280C	17.8	6	167.6	123	3.92	3.440	18.90	1	0	4	4
Merc 450SE	16.4	8	275.8	180	3.07	4.070	17.40	0	0	3	3
Merc 450SL	17.3	8	275.8	180	3.07	3.730	17.60	0	0	3	3
Merc 450SLC	15.2	8	275.8	180	3.07	3.780	18.00	0	0	3	3
Cadillac Fleetwood	10.4	8	472.0	205	2.93	5.250	17.98	0	0	3	4
Lincoln Continental	10.4	8	460.0	215	3.00	5.424	17.82	0	0	3	4
Chrysler Imperial	14.7	8	440.0	230	3.23	5.345	17.42	0	0	3	4
Fiat 128	32.4	4	78.7	66	4.08	2.200	19.47	1	1	4	1
Honda Civic	30.4	4	75.7	52	4.93	1.615	18.52	1	1	4	2
Toyota Corolla	33.9	4	71.1	65	4.22	1.835	19.90	1	1	4	1
Toyota Corona	21.5	4	120.1	97	3.70	2.465	20.01	1	0	3	1
Dodge Challenger	15.5	8	318.0	150	2.76	3.520	16.87	0	0	3	2
AMC Javelin	15.2	8	304.0	150	3.15	3.435	17.30	0	0	3	2
Camaro Z28	13.3	8	350.0	245	3.73	3.840	15.41	0	0	3	4
Pontiac Firebird	19.2	8	400.0	175	3.08	3.845	17.05	0	0	3	2
Fiat X1-9	27.3	4	79.0	66	4.08	1.935	18.90	1	1	4	1
Porsche 914-2	26.0	4	120.3	91	4.43	2.140	16.70	0	1	5	2
Lotus Europa	30.4	4	95.1	113	3.77	1.513	16.90	1	1	5	2
Ford Pantera L	15.8	8	351.0	264	4.22	3.170	14.50	0	1	5	4
Ferrari Dino	19.7	6	145.0	175	3.62	2.770	15.50	0	1	5	6
Maserati Bora	15.0	8	301.0	335	3.54	3.570	14.60	0	1	5	8
Volvo 142E	21.4	4	121.0	109	4.11	2.780	18.60	1	1	4	2

```
print(nycflights13::flights)
```

```
# A tibble: 336,776 x 19
  year month   day dep_time sched_de~1 dep_d~2 arr_t~3 sched~4 arr_d~5 carrier
  <int> <int> <int>   <int>      <int>    <dbl>   <int>   <int>    <dbl> <chr>
1  2013     1     1     517        515         2     830     819      11 UA
2  2013     1     1     533        529         4     850     830     20 UA
3  2013     1     1     542        540         2     923     850     33 AA
4  2013     1     1     544        545        -1    1004    1022    -18 B6
5  2013     1     1     554        600        -6     812     837    -25 DL
6  2013     1     1     554        558        -4     740     728     12 UA
7  2013     1     1     555        600        -5     913     854     19 B6
8  2013     1     1     557        600        -3     709     723    -14 EV
9  2013     1     1     557        600        -3     838     846     -8 B6
10 2013     1     1     558        600        -2     753     745      8 AA
# ... with 336,766 more rows, 9 more variables: flight <int>, tailnum <chr>,
#   origin <chr>, dest <chr>, air_time <dbl>, distance <dbl>, hour <dbl>,
#   minute <dbl>, time_hour <dtm>, and abbreviated variable names
#   1: sched_dep_time, 2: dep_delay, 3: arr_time, 4: sched_arr_time,
#   5: arr_delay
```

Hay varias formas de identificar si un objeto es una tibble o no. La primera es mediante la función `class()`. Por ejemplo:

```
class(mtcars)
```

```
[1] "data.frame"
```

```
class(nycflights13::flights)
```

```
[1] "tbl_df"      "tbl"        "data.frame"
```

Otra forma es fijándonos en la información que nos arroja la función `print()`. Si nos indica el tipo de dato de cada columna es una tibble; además nos imprime al principio que se trata de una tibble indicando las dimensiones de la tabla. También, podemos darnos cuenta de que en las tibbles nos da un número limitado de filas.

## Pregunta 2

Compara y contrasta las siguientes operaciones en el data frame y su equivalente en tibble.

```
df <- data.frame(abc = 1, xyz = "a")
df$x
```

```
[1] "a"
```

```
df[, "xyz"]
```

```
[1] "a"
```

```
df[, c("abc", "xyz")]
```

```
  abc xyz
1   1   a
```

Equivalente en tibble:

```
df <- tibble(abc = 1, xyz = "a")
df$x
```

Warning: Unknown or uninitialised column: `x`.

NULL

```
df[, "xyz"]
```

```
# A tibble: 1 x 1
```

```
  xyz  
  <chr>
```

```
1 a
```

```
df[, c("abc", "xyz")]
```

```
# A tibble: 1 x 2
```

```
  abc xyz  
  <dbl> <chr>
```

```
1      1 a
```

- ¿En qué se parecen?

Se parecen en la forma de acceder a sus elementos mediante corchetes o el símbolo “\$”.

- ¿En qué difieren?

En la forma de mostrar los resultados por pantalla. Las tibbles son más restrictivas; como se puede observar en el código anterior, al no existir una variable llamada `x` cuando accede con `df$x` salta un warning. Con dataframes R interpreta que se trataría de la variable `xyz`.

- ¿Por qué a veces el data frame por defecto nos puede causar mucha frustración?

Al crear subconjuntos empleando corchetes, cuando trabajamos con data frames, en ocasiones nos devolverá un vector y en ocasiones nos devolverá un data frame. Si utilizamos una tibble nos devolverá siempre una tibble.

### Pregunta 3

Si tenemos el nombre de una variable almacenada en un objeto tipo string (por ejemplo `myvar <- "mpg"`), ¿cómo podemos extraer la variable referenciada de una tibble? ¿Y en un data frame?

Tanto en un data frame como en una tibble se podría extraer la variable de la siguiente manera:

```
dataframe[[variable]]
```

### Pregunta 4

Toma la siguiente tibble formada por variables con nombres no sintácticos.

```
df <- tibble(  
  `1` = 1:12,  
  `2` = `1`*2 + `1`*runif(length(`1`))  
)
```

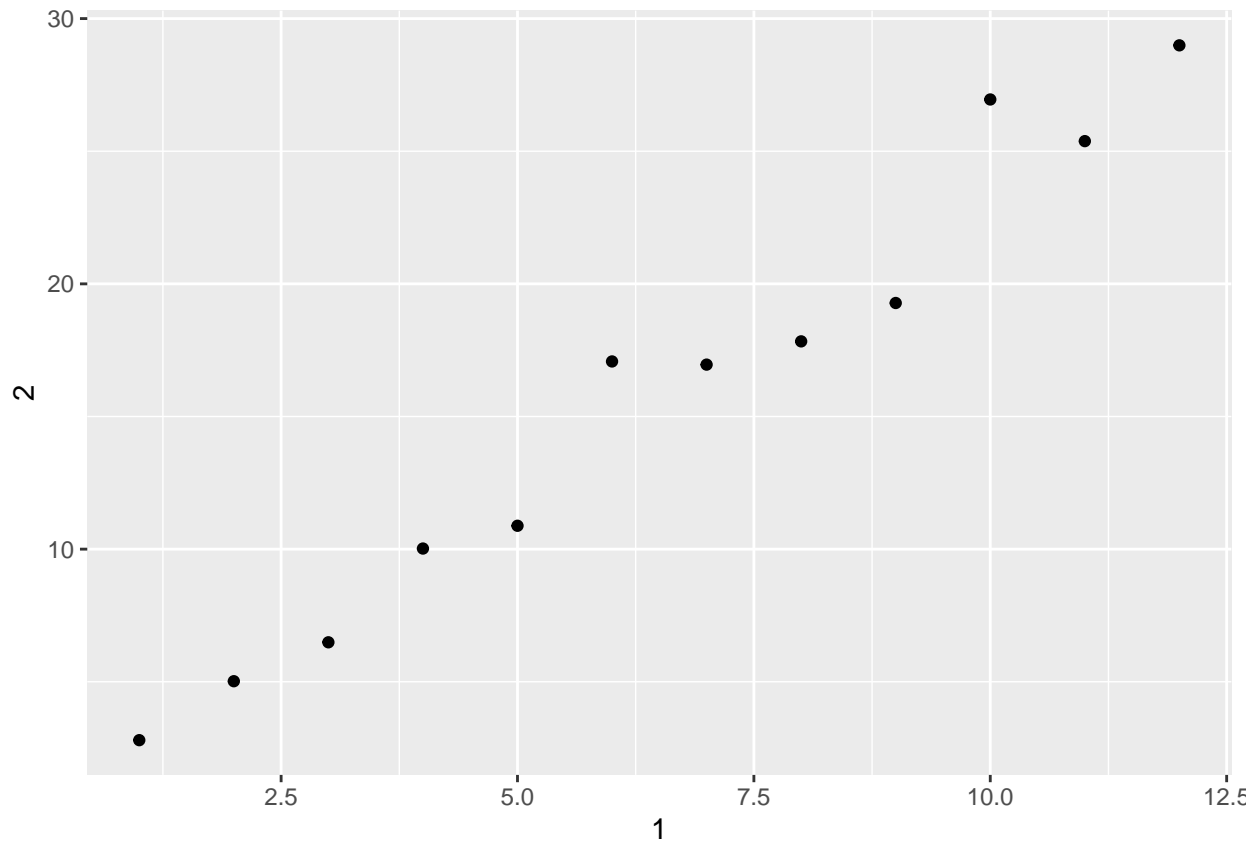
- Extrae el valor de la variable 1

```
df %>% .$`1`
```

```
[1] 1 2 3 4 5 6 7 8 9 10 11 12
```

- Haz un scatterplot de la variable 1 contra la variable 2

```
df %>%  
  ggplot(aes(`1`, `2`)) +  
  geom_point()
```



- Crea una nueva columna llamada 3 que sea el cociente de 2 entre 1.

```
df <- df %>%
  mutate(`3` = `2`/`1`)
```

Renombra las columnas para que se llamen x, y, z respectivamente.

```
df <- df %>%
  rename(
    x = `1`,
    y = `2`,
    z = `3`,
  )
```

- ¿Qué nombre crees que es mejor?

Es mejor de la forma  $x, y, z$  ya que es más fácil extraer el valor de las variables dado que no tienes que tener en cuenta los símbolos cuando números.

## Pregunta 5

Investiga acerca de la función `tibble::enframe()` y `tibble::deframe()`. ¿Qué hace y para qué puede servirte?

`enframe()` convierte vectores atómicos o listas con nombre en data frames de una o dos columnas. Para una lista, el resultado será un tibble anidado con una columna de tipo lista. Para vectores sin nombre, se utiliza la secuencia natural como columna de nombre.

`deframe()` convierte data frames de dos columnas en un vector o lista con nombre, utilizando la primera

columna como nombre y la segunda columna como valor. Si la entrada sólo tiene una columna, se devuelve un vector sin nombre.

## Pregunta 6

¿Cómo podemos controlar cuántos nombres de columna adicionales se imprimen en el footer de una tibble? mediante el argumento `width` de la función `print`. Si queremos que se muestren todas las columnas pondremos `width = Inf`.