Tarea: La carga de Datos con readr

Jesus Mudarra Luján

2023-01-08

Pregunta 1

Indica qué función y parámetros usarías para leer ficheros separados con "|"

Un ejemplo:

```
read_delim("x|y|z\n1|2|3\n4|5|6", delim = "|")
```

Pregunta 2

Además de `file`, `skip` y `comment` que hemos visto en el curso, ¿qué otros argumentos tienen `read_csv` y read_tsv en común? Indica para qué sirve cada uno de ellos.

- `col_names`: `TRUE`, `FALSE` o un vector de caracteres de nombres de columnas. Si es `TRUE`, la primera fila de la entrada se utilizará como nombre de columna y no se incluirá en el marco de datos. Si es `FALSE`, los nombres de las columnas se generarán automáticamente: `X1`, `X2`, `X3`, etc. Si `col_names` es un vector de caracteres, los valores se utilizarán como nombres de las columnas, y la primera fila de la entrada se leerá en la primera fila del marco de datos de salida.
- `col_types`: Si es `NULL`, todos los tipos de columnas se imputarán a partir de las filas guess_max de la entrada intercaladas a lo largo del archivo. Esto es conveniente (y rápido), pero no robusto. Si la imputación falla, necesitará incrementar el guess_max o suministrar los tipos correctos usted mismo.
- `col_select`: Columnas a incluir en los resultados.
- `id`: El nombre de una columna en la que almacenar la ruta del archivo.
- `locale`: La configuración regional controla los valores predeterminados que varían de un lugar a otro.
- `na`: Vector de caracteres de cadenas para interpretar como valores perdidos.
- `quoted_na`: Los valores omitidos entre comillas deben tratarse como valores omitidos (por defecto) o como strings.
- `quote`: Carácter único utilizado para entrecomillar cadenas.
- `trim_ws`: ¿Deben recortarse los espacios en blanco iniciales y finales (espacios ASCII y tabulaciones) de cada campo antes de analizarlo?
- `n_max`: Número máximo de líneas a leer.
- `guess_max`: Número máximo de líneas a utilizar para adivinar los tipos de columna.
- `progress`: ¿Mostrar una barra de progreso? Por defecto sólo se mostrará en una sesión interactiva y no mientras se teje un documento.
- `name_repair` = Tratamiento de los nombres de columna. El comportamiento por defecto es garantizar que los nombres de columna sean `"unique"`.
- `num_threads`: El número de hilos de procesamiento que se utilizarán para el análisis inicial y la lectura perezosa de los datos.

- `show_col_types`: Si es `FALSE`, no mostrar los tipos de columna adivinados. Si es `TRUE` mostrar siempre los tipos de columna, incluso si son suministrados.
- `skip_empty_rows`: ¿Deben ignorarse por completo las filas en blanco? Es decir, si esta opción es `TRUE`, las filas en blanco no se representarán en absoluto. Si es `FALSE`, se representarán con valores `NA` en todas las columnas.
- `lazy`: ¿Leer valores perezosamente? Por defecto, es `FALSE`, porque hay consideraciones especiales cuando se lee un archivo perezosamente que han hecho tropezar a algunos usuarios.

Pregunta 3

Indica los argumentos más importantes de `read_fwf()`

- `col_positions`: Posiciones de columna, creadas por `fwf_empty()`, `fwf_widths()`, `fwf_positions()` o `fwf_cols()`. Para leer sólo los campos seleccionados, se utiliza `fwf_positions()`.
  - `fwf_empty()`: Suposiciones basadas en las posiciones de las columnas vacías.
  - `fwf_widths()`: Proporciona la anchura de las columnas.
  - `fwf_positions()`: Proporciona vectores emparejados de posiciones inicial y final.
  - `fwf_cols()`: Proporciona argumentos con nombre de posiciones de inicio y fin emparejadas o anchuras de columna.

Pregunta 4

A veces un csv contiene necesariamente comas en los campos que son strings. Para evitar problemas en la carga, suelen ir rodeadas de comillas dobles " o de comillas simples '. La convención de `read_csv()` es que asume que cualquier caracter vendrá rodeado por comillas dobles " y si lo queremos cambiar tenemos que usar la función `read_delim()`.

Indica qué argumentos tendríamos que especificar para poder leer el texto del siguiente data frame correctamente

```
"x,y\n1,'a,b'"
```

```
read_delim("x,y\n1,'a,b'", quote = "'")
```

```
Rows: 1 Columns: 2
-- Column specification --------------------------------------------------------
Delimiter: ","
chr (1): y
dbl (1): x

i Use `spec()` to retrieve the full column specification for this data.
i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

```
# A tibble: 1 x 2
      x y
  <dbl> <chr>
1     1 a,b
```

Pregunta 5

Indica qué está mal en la siguiente línea de lectura de CSV

```
read_csv("x,y\n1,2,3\n4,5,6")
```

Falta por incluir el nombre de la tercera columna. Al tener solo dos elementos en la cabecera junta los dos últimos números de las dos filas para que coincida el número de columnas.

Pregunta 6

Indica qué está mal en la siguiente línea de lectura de CSV

```
read_csv("x,y,z\n1,2\n3,4,5,6")
```

En la primera fila no se ha incluido el último valor por lo que se ha añadido un `NA`. Por otro lado, en la segunda fila, al tener 4 elementos y la cabecera 3, se ha juntado el 5 con el 6.

Pregunta 7

Indica qué está mal en la siguiente línea de lectura de CSV

```
read_csv("x,y\n\"1")
read_csv("x,y\n'\',1")
```

El caracter "\" sobra después del "\n". Si se quiere añadir el número 1 como character se debería de añadir como '1'. También, faltaría un segundo elemento en esa fila ya que la cabecera indica que hay dos variables.

Pregunta 8

Indica qué está mal en la siguiente línea de lectura de CSV

```
read_csv("x,y\n1,2\na,b")
```

Se están mezclando tipos de datos. Como en ambas columnas hay dos caracteres `R` considera que las dos columnas son de tipo caracter.

Pregunta 9

Indica qué está mal en la siguiente línea de lectura de CSV

```
read_csv("x;y\n1;2")
```

Los elementos de la tibble se tienen que indicar separados por comas con la función `read_csv()`. Otra opción es usar la función `read_csv2()` si se quiere tener los elementos separados por ;.