Herramientas de edición de datos I

2024

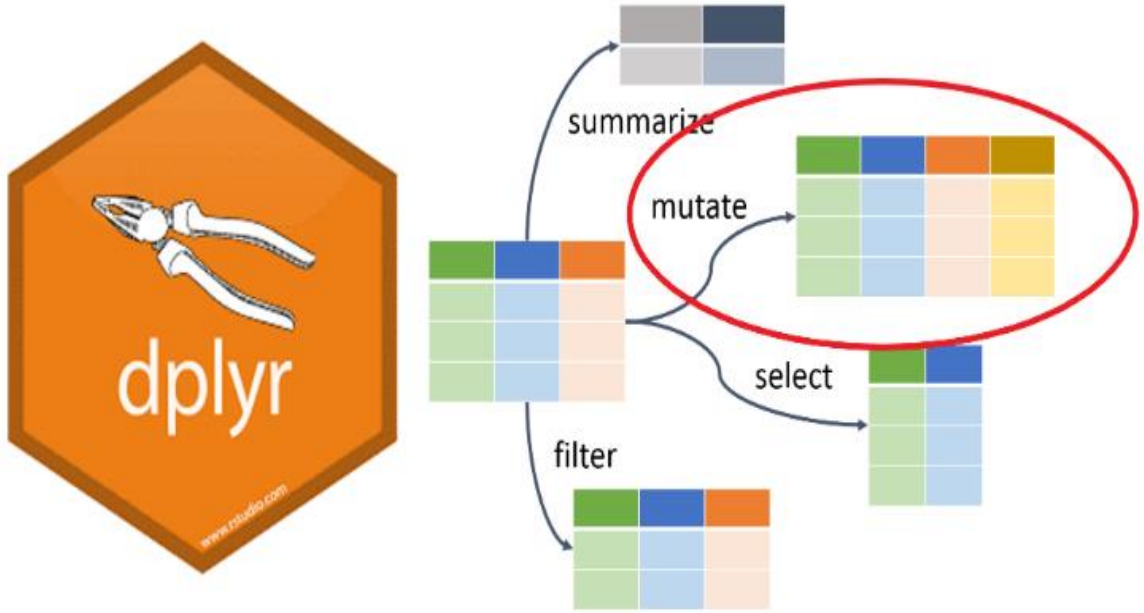
Contenidos

01. Edición de datos I

Trabajando con la BBDD...

El siguiente paso es modificar columnas existentes o agregarle nuevas columnas en base a información existente o externa.

Para eso tenemos la función **mutate.**



Su sintaxis es la siguiente:

```
mutate(data.frame, nombre_columna1 = expresion1,
nombre_columna2 = expresion2, ...)
```

02. Mutate

Crear columna nueva

- Para ilustrar la creación de columnas, creemos un dataframe de ejemplo:

```
compras = tibble(cantidad = c('4', '2', '5'),
                 precio = c(200, 300, 400))
```

```
compras
```

```
## # A tibble: 3 × 2
##   cantidad precio
##   <chr>     <dbl>
## 1 4           200
## 2 2           300
## 3 5           400
```

Queremos crear una columna llamada valor_total, que sea la multiplicación de las dos columnas:

```
compras_completo = compras %>% mutate(valor_total = cantidad * precio)
```

¿Qué pasó?

```
Error in `mutate()`:
i In argument: `valor_total = cantidad * precio`.
Caused by error in `cantidad * precio`:
! non-numeric argument to binary operator
Backtrace:
 1. compras %>% mutate(valor_total = cantidad * precio)
 3. dplyr:::mutate.data.frame(., valor_total = cantidad * precio)
 4. dplyr:::mutate_cols(.data, dplyr_quosures(...), by)
 6. dplyr:::mutate_col(dots[[i]], data, mask, new_columns)
 7. mask$eval_all_mutate(quo)
 8. dplyr (local) eval()

Error in mutate(., valor_total = cantidad * precio) :
 Caused by error in `cantidad * precio`:
 ! non-numeric argument to binary operator
```

Reemplazar columna existente

Primero debemos asegurarnos que ambas columnas sean numéricas:

```
class(compras$cantidad)
```

```
## [1] "character"
```

```
class(compras$precio)
```

```
## [1] "numeric"
```

Reemplazar columna existente

Transformamos cantidad en numérica, para habilitar operaciones de este tipo:

```
compras_numeric = compras %>% mutate(cantidad = as.numeric(cantidad))
```

```
class(compras_numeric$cantidad)
```

```
## [1] "numeric"
```

```
class(compras_numeric$precio)
```

```
## [1] "numeric"
```

Crear columna nueva

Volvemos a nuestro ejemplo original:

```
compras_completo = compras_numeric %>% mutate(valor_total = cantidad * precio)
```

cantidad	precio	valor_total
4	200	800
2	300	600
5	400	2000

¿Qué otras cosas podemos hacer?

```
compras_mas_completo = compras_completo %>% mutate(
    valor_cte = 10,
    max_precio = max(precio),
    min_precio = min(precio),
    id_fila = row_number(),
    col_modificada = precio - 100,
    num_digitos = valor_total %>% as.character %>% stringr::str_length()
```

cantidad	precio	valor_total	valor_cte	max_precio	min_precio	id_fila	col_modificada	num_digitos
4	200	800	10	400	200	1	100	3
2	300	600	10	400	200	2	200	3
5	400	2000	10	400	200	3	300	4

¿Qué otras cosas podemos hacer?

- En resumen, de alguna forma u otra, podemos realizar cualquier modificación que se nos ocurra a las columnas de la tabla.
- Las opciones son cientos, por lo que la recomendación es: "Ante la duda, googlear"

Anexo

Funciones clásicas para transformación de columnas:

- **as.numeric** pasa variable a formato numérico. Filas deben contener solo números para que pueda utilizarse.
- **as.character** pasa variable a formato texto.
- **as.factor** pasa variable a formato factor, práctico en gráficos y eficiente en términos de memoria.
- **as.Date** pasa variable a formato fecha. Filas deben cumplir con formato específico para que puedan ser interpretadas como fechas.
- **toupper / tolower:** pasa columna string a mayúsculas/minúsculas, respectivamente.

Anexo

Funciones clásicas para transformación de columnas:

- **replace_NA:** reemplaza valores nulos (NA) con el valor especificado.
- **round:** redonea columna numérica según la precisión que se ingrese. Por defecto, deja en valor entero un número decimal.
- **sum/mean/max/min/sd/median:** Obtiene la suma, media, máximo, mínimo, desviación estándar y mediana, respectivamente, de la columna numérica especificada.
- **paste:** concatena los valores de dos columnas string, separadas por un espacio.

¡Muchas gracias!

Instituto Nacional de Estadísticas · Chile

www.ine.gob.cl