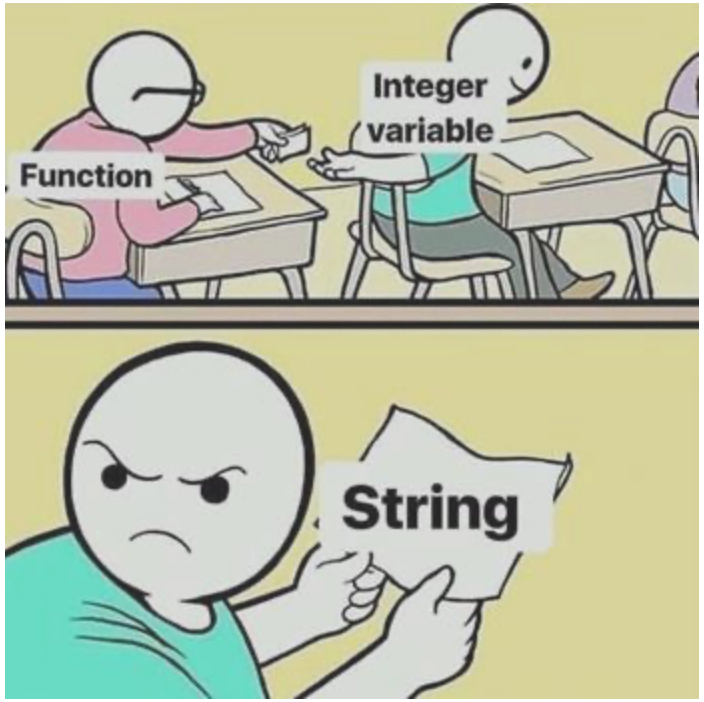


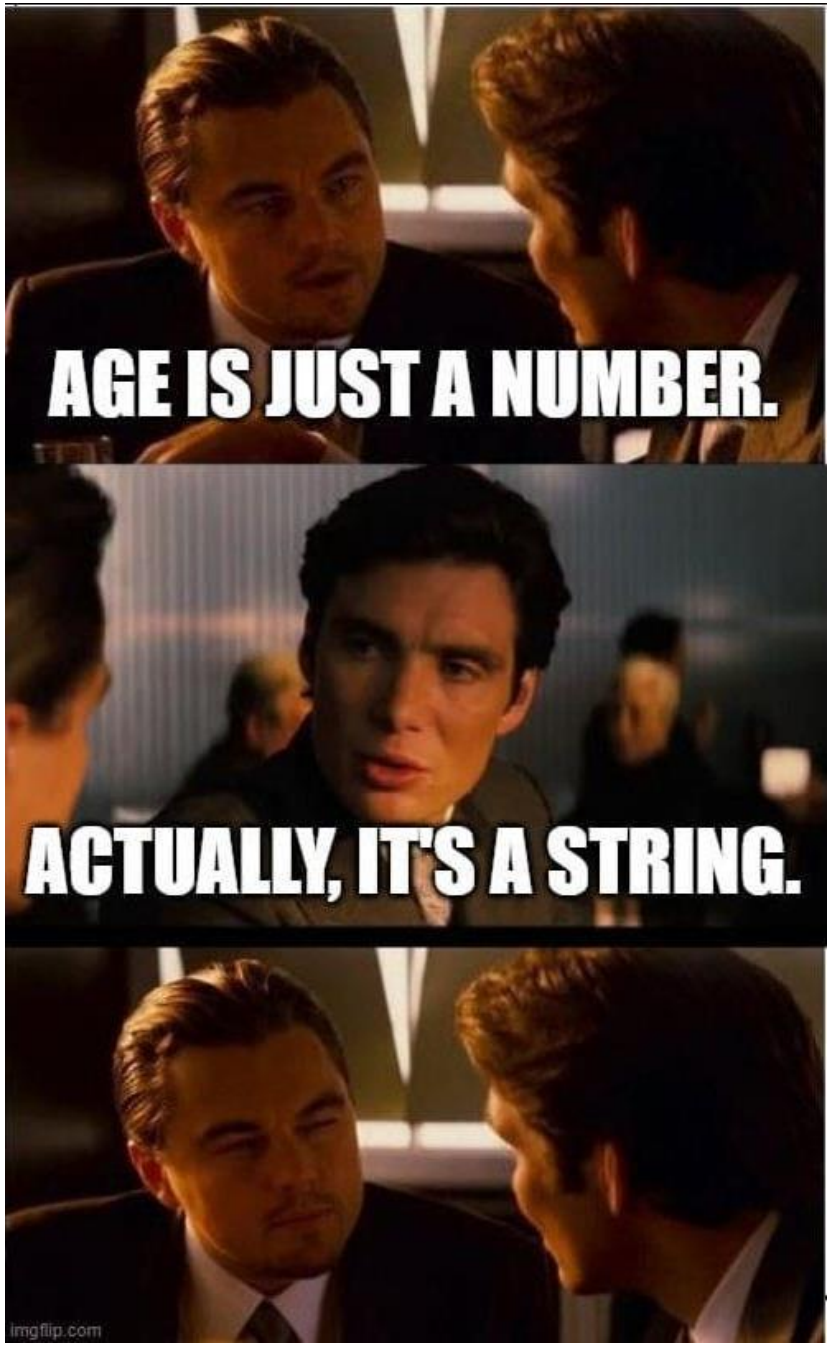
R: datos de texto

{stringr}

José Luis Texcalac Sangrador

Procesamiento y visualización de datos espaciales en R

INSP/ESPM
Escuela de Salud Pública de México



str_c(..., sep = "")

str_c

Permite concatenar (unir) el texto de dos o más columnas en una sola.

```
malla %>%
  mutate(folio = str_c(posgrado, matricula, sexo)) %>%
  print()
```

matricula	sexo	Posgrado	folio
0718	1	MCSA	MCSA07181
2183	2	MCEP	MCEP21832
1241	1	MCSS	MCSS12411
0213	2	MCSA	MCSA02132
4386	2	MCBI	MCBI43862

str_c(..., sep = "")

```
dataset %>%
  mutate(folio = str_c(posgrado,
                       matricula,
                       sexo,
                        sep = "-")) %>%
  print()
```

matricula	sexo	Posgrado	folio
0718	1	MCSA	MCSA-0718-1
2183	2	MCEP	MCEP-2183-2
1241	1	MCSS	MCSS-1241-1
0213	2	MCSA	MCSA-0213-2
4386	2	MCBI	MCBI-4386-2

sep = “”

Permite agregar un separador entre los elementos a concatenar (por ejemplo, un guión)

1

001

Vector o variable a “inflar”

De qué lado se pone el texto que va a “inflar” (por defecto a la izquierda)

```
str_pad(string, width, side = c("left", "right", "both"), pad = " ")
```

Longitud que debe tener el texto

Texto con el que se va a “inflar”

```
malla %>%
  mutate(clave_mun = str_pad(mun, 3, pad = "0"))
```

mun	municipio	clave_mun
1	San Juan	001
72	Santa María	072
348	Guadalupe	348
9	San Pablo	009

Concatenar `str_c()` y `str_pad()`

```
malla %>%
   mutate(folio = str_c(id, nombre))
```

id	nombre	folio
1	Juan	1Juan
2	Pedro	2Pedro
3	Felipe	3Felipe

```
malla %>%
   mutate(folio = str_c(id, nombre, sep = "-"))
```

id	nombre	folio
1	Juan	1-Juan
2	Pedro	2-Pedro
3	Felipe	3-Felipe

```
malla %>%
   mutate(folio = str_c(str_pad(id, 2, pad = "0"),
                        nombre,
                        sep = "-"))
```

id	nombre	folio
1	Juan	01-Juan
2	Pedro	02-Pedro
3	Felipe	03-Felipe

str_sub(texto, start = 1, end = 4)

```
malla %>%
  mutate(posgrado = str_sub(folio,
                            start = 1,
                            end = 4)) %>%
  print()
```

matricula	sexo	folio	posgrado
0718	1	MCSA-0718-1	MCSA
2183	2	MCEP-2183-2	MCEP
1241	1	MCSS-1241-1	MCSS
0213	2	MCSA-0213-2	MCSA
4386	2	MCBI-4386-2	MCBI

str_sub

Permite extraer texto de un vector.

start =

En el ejemplo, el número 1 indica que se iniciará a extraer el texto a partir del primer carácter.

end =

En el ejemplo, el número 4 indica que se finalizará la extracción de texto en el cuarto carácter.

str_sub(texto, start = 6L)

```
malla %>%
  mutate(matri_sexo = str_sub(folio,
                              start = 6)) %>%
  print()
```

matricula	sexo	folio	matri_sexo
0718	1	MCSA-0718-1	0718-1
2183	2	MCEP-2183-2	2183-2
1241	1	MCSS-1241-1	1241-1
0213	2	MCSA-0213-2	0213-2
4386	2	MCBI-4386-2	4386-2

str_sub

Permite extraer texto de un vector.

start =

En el ejemplo, el número 5 indica que se iniciará a extraer el texto a partir del quinto carácter.

end =

Si no se indica este argumento entonces el texto restante a partir del carácter de inicio es seleccionado.


```
sun <-
    read_csv("./data/Base_SUN_2018.csv") %>%
    clean_names() %>%
    print()
```

```
Rows: 1089 Columns: 9
```

```
── Column specification ────────────────────────────────────────────────────────
Delimiter: ","
chr (8): CVE_ENT, NOM_ENT, CVE_MUN, NOM_MUN, CVE_LOC, NOM_LOC, CVE_SUN, NOM_SUN
dbl (1): POB_2018

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: 1,089 × 9
   cve_ent nom_ent             cve_mun nom_mun              cve_loc nom_loc cve_sun nom_sun     pob_2018
   <chr>   <chr>               <chr>   <chr>                <chr>   <chr>   <chr>   <chr>          <dbl>
 1 01      Aguascalientes      01011   "San Francisco de …  NA      NA      M01.01  Aguascalie…    42531
 2 01      Aguascalientes      01005   "Jes\xfas Mar\xeda"  NA      NA      M01.01  Aguascalie…   116700
 3 01      Aguascalientes      01001   "Aguascalientes"     NA      NA      M01.01  Aguascalie…   897331
 4 02      Baja California     02005   "Playas de Rosarit…  NA      NA      M02.03  Tijuana       110683
 5 02      Baja California     02003   "Tecate"             NA      NA      M02.03  Tijuana       115570
 6 02      Baja California     02004   "Tijuana"            NA      NA      M02.03  Tijuana      1798741
 7 02      Baja California     02002   "Mexicali"           NA      NA      M02.02  Mexicali     1065882
 8 02      Baja California     02001   "Ensenada"           NA      NA      M02.01  Ensenada      542896
 9 03      Baja California Sur 03003   "La Paz"             NA      NA      M03.01  La Paz        313204
10 04      Campeche            04002   "Campeche"           NA      NA      M04.01  Campeche      298741
# … with 1,079 more rows
```

Averiguamos la codificación que mejor se adapte a nuestra malla de datos

```
guess_encoding("./ruta/file.csv", n_max = 1000)
```

```
# A tibble: 2 × 2
  encoding   confidence
  <chr>           <dbl>
1 ISO-8859-1       0.36
2 ISO-8859-2       0.26
```

Importamos la malla con la codificación recomendada

```
read_csv("./ruta/file.csv", locale = readr::locale(encoding = "ISO-8859-1"))
```

```
# A tibble: 1,089 × 9
   cve_ent nom_ent          cve_mun nom_mun            cve_loc nom_loc cve_sun nom_sun     pob_2018
   <chr>   <chr>            <chr>   <chr>              <chr>   <chr>   <chr>   <chr>          <dbl>
 1 01      Aguascalientes   01011   San Francisco de l… NA      NA      M01.01  Aguascalie…    42531
 2 01      Aguascalientes   01005   Jesús María         NA      NA      M01.01  Aguascalie…   116700
 3 01      Aguascalientes   01001   Aguascalientes      NA      NA      M01.01  Aguascalie…   897331
 4 02      Baja California  02005   Playas de Rosarito  NA      NA      M02.03  Tijuana       110683
 5 02      Baja California  02003   Tecate              NA      NA      M02.03  Tijuana       115570
```

Generar variables condicionales case_when( )

Generar nueva columna

Columna con los valores a identificar

Condiciones
- mayor qué
- menor qué
- igual a

```
malla %>%
  mutate(new_var = case_when(variable == "condición_1" ~ "resultado",
                             variable == "condición_2" ~ "resultado",
                             TRUE ~ variable))
```

Valor de la condición
- número
- texto
- lógico

Valor a almacenar en la nueva variable

x < y	menor qué
x > y	mayor qué
x == y	igual a
x <= y	menor o igual a
x >= y	mayor o igual a
x != y	diferente de
x %in% y	pertenece a
is.na(x)	is NA
!is.na(x)	Distinto de NA

Generar variables condicionales case_when( )

```
malla
```

nombre	edad
Luisa	27
Juana	31
Petra	28
María	41
Andrea	33

```
malla %>%
  mutate(edad_gpo = case_when(edad >= 20 & edad < 30 ~ "20 a 29",
                              edad >= 30 & edad < 40 ~ "30 a 39",
                              edad >= 40 ~ "40 y más")) %>%
  print()
```

nombre	edad	edad_gpo
Luisa	27	"20 a 29"
Juana	31	"30 a 39"
Petra	28	"20 a 29"
María	41	"40 y más"
Andrea	33	"30 a 39"

Generar variables condicionales case_when( )

```
malla
```

nombre	edad
Luisa	27
Juana	31
Petra	28
María	41
Andrea	33

```
malla %>%
  mutate(edad_gpo = case_when(edad >= 20 & edad < 30 ~ 2L,
                              edad >= 30 & edad < 40 ~ 3L,
                              edad >= 40 ~ 4L)) %>%
  print()
```

nombre	edad	edad_gpo
Luisa	27	2
Juana	31	3
Petra	28	2
María	41	4
Andrea	33	3

Recodificar valores

Activar paquete en la sesión

Comando para convertir valores a NA

Columna que contiene los valores NA

```
library(fauxnaif)
malla %>%
  mutate(sexo = na_if_in(sexo, 98, 99),
         edad = na_if_in(edad, -88, -99),
         peso = na_if_in(peso, -88, -99)) %>%
  print()
```

Genero columna, si ya existe entonces la sobreescribe.

Valores a convertir a NA

```
malla
```

nombre	sexo	edad	peso
Kevin	1	17	-99
Brayan	1	-99	61.7
Kimberly	98	15	51.9
Britany	2	16	59.3
Brandon	99	17	-88
Melany	2	-88	61.6

nombre	sexo	edad	peso
Kevin	1	17	NA
Brayan	1	NA	61.7
Kimberly	NA	15	51.9
Britany	2	16	59.3
Brandon	NA	17	NA
Melany	2	NA	61.6

Recodificar valores

Es posible hacer todo el proceso para varias columnas a la vez

```
malla
```

nombre	sexo	edad	peso
Kevin	1	17	-99
Brayan	1	-99	61.7
Kimberly	98	15	51.9
Britany	2	16	59.3
Brandon	99	17	-88
Melany	2	-88	61.6

```
library(fauxnaif)
malla %>%
  mutate(across(sexo:peso, ~ na_if_in(., -88, -99, 98, 99)) %>%
  print()
```

nombre	sexo	edad	peso
Kevin	1	17	NA
Brayan	1	NA	61.7
Kimberly	NA	15	51.9
Britany	2	16	59.3
Brandon	NA	17	NA
Melany	2	NA	61.6

Trabajar con NA

nombre	edad	peso	pelo
juan	18	68.3	lacio
eva	NA	70.1	NA
luis	19	69.4	lacio
ana	20	NA	lacio
mario	20	73.5	chino
edith	19	65.2	NA
david	21	76.4	NA

```
malla %>% drop_na()
malla %>% na_omit()
```

nombre	edad	peso	pelo
juan	18	68.3	lacio
luis	19	69.4	lacio
mario	20	73.5	chino

Trabajar con NA

nombre	edad	peso	pelo
juan	18	68.3	lacio
eva	NA	70.1	NA
luis	19	69.4	lacio
ana	20	NA	lacio
mario	20	73.5	chino
edith	19	65.2	NA
david	21	76.4	NA

```
malla %>% drop_na(pelo)
```

nombre	edad	peso	pelo
juan	18	68.3	lacio
luis	19	69.4	lacio
ana	20	NA	lacio
mario	20	73.5	chino

Trabajar con NA

nombre	edad	peso	pelo
juan	18	68.3	lacio
eva	NA	70.1	NA
luis	19	69.4	lacio
ana	20	NA	lacio
mario	20	73.5	chino
edith	19	65.2	NA
david	21	76.4	NA

```
malla %>% drop_na(edad, peso)
```

nombre	edad	peso	pelo
juan	18	68.3	lacio
luis	19	69.4	lacio
mario	20	73.5	chino
edith	19	65.2	NA
david	21	76.4	NA