



Manipulación de caracteres R

José Luis Texcalac Sangrador

Procesamiento y visualización de datos espaciales en R

INSP-ESPM



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

```
mallá %>% drop_na()
```

```
mallá %>% 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

```
mallat %>% 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

```
mallat %>% 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



```
str_c(..., sep = "")
```

str_c

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

```
dataset %>%  
  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()
```

`sep = ""`

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

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



Su turno...

- Cargue la malla `cov_def`.
- Concatene las columnas de entidad y municipio, nombre a la nueva columna como `cvegeo`.
- Mueva la nueva columna a la primera posición de la malla.
- Renombre columnas de entidad y municipio de residencia como `cve_ent` y `cve_mun`.
- Guarde el resultado en la misma malla.

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"

```
mallá %>%
```

```
  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 `c()` y `str_pad()`

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

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

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

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

```
mallá %>%  
  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 = 1L, end = -1L)
```

```
mallá %>%  
  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 = 5L)`

```
mall %>%  
  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.



Su turno...

- Trabaje con la malla `cov_def`.
- Use el comando `str_sub` y genere columnas `year`, `month` y `day` a partir de la columna `fecha_def`.
- Posicione las nuevas columnas después de `fecha_def`.
- Guarde el resultado en la misma malla.
- Genere una nueva malla que contabilice el número de defunciones por municipio, guarde el resultado como `def_mun`.

Unión de mallas de datos



inner_join()

`inner_join(x, y)`

1	x1	1	y1
2	x2	2	y2
3	x3	4	y4



```
inner_join(malla_1, malla_2, by = "cvegeo")
```

Tengo la tabla de municipios
y requiero de su clave SUN

malla_1

cvegeo	nom_mun
17007	Cuernavaca
17020	Tepoztlán
17028	Xochitepec

Tengo la tabla del Sistema
Urbano Nacional

malla_2

cvegeo	cve_sun
17009	M17.02
17011	M17.02
17020	M17.02

El comando `inner_join` genera una nueva tabla que combina únicamente la información de los **identificadores** coincidentes

cvegeo	nom_mun	cve_sun
17020	Tepoztlán	M17.02

```
inner_join(malla_1, malla_2, by = "cvegeo")
```

malla_1

<i>cvegeo</i>	<i>nom_mun</i>
17017	Puente de Ixtla
17020	Tepoztlán
17028	Xochitepec

malla_2

<i>cvegeo</i>	<i>cve_sun</i>
17009	M17.02
17011	M17.02
17017	P17.01
17017	C17.02

<i>cvegeo</i>	<i>nom_mun</i>	<i>cve_sun</i>
17017	Puente de Ixtla	P17.01
17017	Puente de Ixtla	C17.02



Argumento by

- Si el nombre del identificador de ambas columnas no coincide entonces:

X Y

```
by = c("edo" = "estado")
```

En la unión se mantiene el nombre de la columna de la tabla X

- Si la unión es por más de un identificador entonces:

```
by = c("edo" = "estado", "mun" = "mun")
```



```
inner_join(malla_1, malla_2, by = c("cvegeo" = "cve_mun"))
```

malla_1

<i>cvegeo</i>	<i>nom_mun</i>
17017	Puente de Ixtla
17020	Tepoztlán
17028	Xochitepec

malla_2

<i>cve_mun</i>	<i>cve_sun</i>
17009	M17.02
17011	M17.02
17017	P17.01
17017	C17.02

<i>cvegeo</i>	<i>nom_mun</i>	<i>cve_sun</i>
17017	Puente de Ixtla	P17.01
17017	Puente de Ixtla	C17.02



left_join()

`left_join(x, y)`

1	x1	1	y1
2	x2	2	y2
3	x3	4	y4

```
left_join(malla_1, malla_2, by = "cvegeo")
```

malla_1

cvegeo	nom_mun
17007	Cuernavaca
17020	Tepoztlán
17028	Xochitepec

malla_2

cvegeo	cve_sun
17009	M17.02
17011	M17.02
17020	M17.02

cvegeo	nom_mun	cve_sun
17007	Cuernavaca	NA
17020	Tepoztlán	M17.02
17028	Xochitepec	NA

right_join()

`right_join(x, y)`

1	x1	1	y1
2	x2	2	y2
3	x3	4	y4

```
right_join(malla_1, malla_2, by = "cvegeo")
```

malla_1

<i>cvegeo</i>	<i>nom_mun</i>
17007	Cuernavaca
17020	Tepoztlán
17028	Xochitepec

malla_2

<i>cvegeo</i>	<i>cve_sun</i>
17009	M17.02
17011	M17.02
17020	M17.02

<i>cvegeo</i>	<i>nom_mun</i>	<i>cve_sun</i>
17009	NA	M17.02
17011	NA	M17.02
17020	Tepoztlán	M17.02



full_join()

full_join(x, y)

1	x1	1	y1
2	x2	2	y2
3	x3	4	y4



```
full_join(malla_1, malla_2, by = "cvegeo")
```

malla_1

<i>cvegeo</i>	<i>cve_mun</i>
17007	Cuernavaca
17020	Tepoztlán
17028	Xochitepec

malla_2

<i>cvegeo</i>	<i>cve_sun</i>
17009	M17.02
17011	M17.02
17020	M17.02

<i>cvegeo</i>	<i>nom_mun</i>	<i>cve_sun</i>
17007	Cuernavaca	NA
17009	NA	M17.02
17011	NA	M17.02
17020	Tepoztlán	M17.02
17028	Xochitepec	NA



```
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



Su turno...

- Importe la malla `Base_SUN_2018.csv` con la codificación adecuada, nombre a su objeto como `sun`.
 - Seleccione sólo Zonas Metropolitanas
 - Genere malla con la población por ZM, guarde el resultado como `pob_sun`.
- Trabaje con la malla `cov_def` y seleccione las columnas: `cvegeo`, `cve_ent`, `cve_mun`, `fecha_def`, guarde el resultado como `def_sun`.
- Agregue las columnas `cve_sun` y `nom_sun` a la malla `def_sun` (conservar sólo los municipios que pertenecen a alguna Zona Metropolitana).
 - ¿cuántas defunciones pertenecen a municipios de zonas metropolitanas?
- Agregue las defunciones diarias por zona metropolitana (guarde el resultado como `def_sun`).
- Filtre de la malla `def_sun` los registros de las 5 zonas metropolitanas con mayor número de defunciones.
- Genere un gráfico con la malla `def_sun` en el que se vea la evolución diaria del total de defunciones por Zona Metropolitana, guarde el gráfico como `g_def_sun`.



Otros tipos de graficación

```
g_def_sun <-  
  ggplot(data = def_sun) +  
  geom_line(aes(x = fecha_def, y = tot_def, colour = nom_sun)) +  
  scale_color_brewer(palette = "Set2") +  
  theme_minimal()  
  
g_def_sun
```

```
remotes::install_github("plotly/plotly")
```

```
library(plotly)  
ggplotly(g_def_sun)
```



Su turno...

- Importe la malla `Base_SUN_2018.csv` con la codificación adecuada, nombre a su objeto como `sun`.
- Filtre las zonas metropolitanas de la malla `sun`.
- Genere una malla solo con los municipios que pertenecen a alguna zona metropolitana, la malla debe contener el nombre de la zona, la clave sun, el nombre del municipio y el total de defunciones (guarde el resultado como `def_mun_sun`).
 - ¿Cuántos municipios pertenecen a alguna Zona Metropolitana?
 - ¿Cuál es la Zona Metropolitana con mayor número de defunciones? (guarde el resultado como `def_mun`).
 - ¿Cuál fue el municipio con mayor número de defunciones?