

**lubridate**

**Entornos de Análisis de Datos: R**

**Alberto Torres Barrán**

**2021-01-07**

# Fechas

- El paquete `lubridate` implementa dos tipos principales de fechas/horas:
  - Fechas, `<date>`
  - Fecha-hora, `<dtm>` (*timestamps*)
- Se pueden crear a partir de:
  - cadenas de caracteres: `ymd()`, `mdy()`, `dmy()`, `ymd_hms()`, etc.
  - componentes: `make_date` y `make_datetime()`
  - otros objetos que representan fechas: `as_datetime()`, `as_date()`

# Creación a partir de cadenas

- El nombre de la función representa el orden de los componentes de la fecha:

```
ymd("20181012")  
## [1] "2018-10-12"
```

```
ydm("20181012")  
## [1] "2018-12-10"
```

- Mientras el orden sea correcto, se reconocen también formatos específicos de la localización

```
sys.getlocale("LC_TIME")  
## [1] "Spanish_Spain.1252"
```

```
dmy("20 de Enero de 2017")  
## [1] "2017-01-20"
```

- Así como fechas en inglés

```
mdy("January 5th 2017")  
## [1] "2017-01-05"
```

# Más ejemplos

- Todas las funciones están vectorizadas (aplican sobre vectores elemento a elemento)

```
ymd(c('20200401', '2020402', '20200403'))  
## [1] "2020-04-01" NA "2020-04-03"
```

- Si un elemento del vector representa una fecha incorrecta se cambia por NA

# Creación a partir de sus componentes

- `make_date` se usa para crear fechas a partir de los componentes por separado

```
make_date(2015, 10, 20)  
## [1] "2015-10-20"
```

- Los componentes pueden ser numéricos o cadenas de caracteres

```
make_date('2015', 10, 20)  
## [1] "2015-10-20"
```

- `make_date` y `make_datetime` también aceptan vectores

```
make_date(2020, c(1, 2, 4), 5:7)  
## [1] "2020-01-05" "2020-02-06" "2020-04-07"
```

# Funciones `as_date` y `as_datetime`

- Útiles para convertir otros formatos de fechas
- El ejemplo más común son fechas/horas codificadas en formato "Unix epoch" (segundos desde el 1 de Enero de 1970)

```
as_datetime(c(1587399491, 1587399492))  
## [1] "2020-04-20 16:18:11 UTC" "2020-04-20 16:18:12 UTC"
```

# R base

- Es importante no confundir las clases del paquete `lubridate` con las que existen en R base:
  1. clase `Date` , para fechas
  2. clases `POSIXct` , `POSIXlt` para fechas y horas
- Se recomienda limitarse a las funciones de la librería `lubridate`
- Por tanto, evitar usar funciones de R base como
  1. `as.Date`
  2. `as.POSIXct`
  3. `strftime`
  4. `strptime`

# Obtener componentes de una fecha

- `year()` , año
- `month()` , mes
- `mday()` , día del mes; `wday()` , día de la semana; `yday()` , día del año
- `hour()` , hora
- `minute()` , minuto
- `second()` , segundo



# Ejemplos

- Día del mes

```
fecha <- ymd(c("20181205", "20181208", "20181210", "20181220"))
mday(fecha)
## [1] 5 8 10 20
```

- Día de la semana

```
wday(fecha)
## [1] 4 7 2 5
```

- Nombre de los días, empezando en lunes

```
wday(fecha, label = TRUE, abbr = FALSE, week_start = 1)
## [1] miércoles sábado lunes jueves
## 7 Levels: lunes < martes < miércoles < jueves < viernes < ... < domingo
```

- Asignar un nuevo valor

```
hour(fecha) <- 18
fecha
## [1] "2018-12-05 18:00:00 UTC" "2018-12-08 18:00:00 UTC"
## [3] "2018-12-10 18:00:00 UTC" "2018-12-20 18:00:00 UTC"
```

# Aritmética con fechas

- lubridate implementa las funciones hours, days, months, etc. para representar períodos
- Con estas funciones podemos hacer operaciones con fechas:

```
ymd("2011-01-01") + years(1)
## [1] "2012-01-01"
ymd("2020-03-01") + months(3)
## [1] "2020-06-01"

# 5 reuniones semanales, empezando el 1 de abril de 2020
ymd("2020-04-01") + weeks(0:4)
## [1] "2020-04-01" "2020-04-08" "2020-04-15" "2020-04-22" "2020-04-29"

today()
## [1] "2021-01-07"
today() + days(15)
## [1] "2021-01-22"
```

# Operadores especiales

- Al realizar una operación, si la fecha no existe `lubridate` devuelve `NA`

```
ymd("2013-01-31") + months(0:4)
## [1] "2013-01-31" NA "2013-03-31" NA "2013-05-31"
```

- Existen operadores especiales que devuelven la última fecha válida del mes

```
# suma
ymd("2013-01-31") %m+% months(0:4)
## [1] "2013-01-31" "2013-02-28" "2013-03-31" "2013-04-30" "2013-05-31"
```

```
# resta
ymd("2013-01-31") %m-% months(0:4)
## [1] "2013-01-31" "2012-12-31" "2012-11-30" "2012-10-31" "2012-09-30"
```

# Redondear fechas

- `round_date()`
- `floor_date()` (hacia abajo)
- `ceiling_date()` (hacia arriba)

```
fecha <- ymd(c("20181205", "20181209", "20181210", "20181220"))
floor_date(fecha, "week")
## [1] "2018-12-02" "2018-12-09" "2018-12-09" "2018-12-16"
```

```
fecha <- ymd(c("20181205", "20181209", "20181210", "20181220"))
floor_date(fecha, "week", week_start = 1)
## [1] "2018-12-03" "2018-12-03" "2018-12-10" "2018-12-17"
```