

Manipulación de Fechas en R

R con Enfoque Financiero: AulaABA

2023-05-13

La manipulación de fechas es una tarea común en el análisis de datos, y R ofrece diferentes herramientas para realizar esta tarea. En particular, las librerías base de R y lubridate son muy útiles para la manipulación de fechas.

En la librería base de R, las fechas se almacenan como objetos de la clase “Date” o “POSIXct”. Para manipular fechas en R, se pueden utilizar las funciones de la librería base como `strptime()` para convertir cadenas de caracteres en objetos de fecha, `format()` para cambiar el formato de una fecha, y `difftime()` para calcular la diferencia entre dos fechas.

Por otro lado, la librería lubridate ofrece una serie de funciones para la manipulación de fechas de manera más intuitiva. Algunas de las funciones más comunes de lubridate son:

- `ymd()`: Esta función convierte una cadena de caracteres en una fecha en formato “año-mes-día”.
- `hms()`: Esta función convierte una cadena de caracteres en una hora en formato “horas-minutos-segundos”.
- `month()`: Esta función extrae el mes de una fecha.
- `year()`: Esta función extrae el año de una fecha.
- `day()`: Esta función extrae el día de una fecha.
- `hour()`: Esta función extrae la hora de una fecha.
- `minute()`: Esta función extrae los minutos de una fecha.
- `second()`: Esta función extrae los segundos de una fecha.

Además, lubridate también ofrece funciones para realizar operaciones aritméticas con fechas, como por ejemplo, sumar o restar días, semanas, meses o años a una fecha determinada.

```
suppressWarnings(suppressPackageStartupMessages(library(tidyverse)))
suppressWarnings(suppressPackageStartupMessages(library(lubridate)))
##Importacion de datos
letras_bc <- read.table("../ ../ ../data/letras_bc_consolidado_clean.csv")

##Creamos una columna de FechaSubasta utilizando una
#secuencia de fechas de la fecha minima de subasta
#de nuestro dataframe original hasta la fecha maxima por dia

##Creamos una segunda columna que aleatoriamente
#nos dira 1 si fue declarado desierto y
#0 si no fue declarada desierta
subasta_desierta <- data.frame(
  "FechadeSubasta" = seq(as.Date(min(letras_bc$FechadeSubasta)),
                        as.Date(max(letras_bc$FechadeSubasta)), by="day" ),
  "DeclaradaDesierta"= sample(c(1,0),length(seq(as.Date(min(letras_bc$FechadeSubasta)),
as.Date(max(letras_bc$FechadeSubasta)), by="day" )), TRUE))
```

```
##Transformamos la columna fecha de subasta
#en formato de fecha ya que estaba en caracter

letras_bc$FechadeSubasta <- as.Date(letras_bc$FechadeSubasta)

##Haciendo left join
union_de_desiertas <- left_join(letras_bc, subasta_desierta, by="FechadeSubasta")
glimpse(union_de_desiertas)
```

```
## Rows: 536
## Columns: 7
## $ FechadeSubasta      <date> 2007-04-03, 2010-02-17, 2010-02-24, 2023-04-12, 201~
## $ FechaLiquidacion    <chr> "2007-04-04", "2010-02-02", "2010-02-02", "2023-04-0~
## $ MontoSubastado      <int> 400, 300, 500, 5000, 2500, 2500, 500, 800, 400, 750,~
## $ MontoDemandado      <dbl> 2281.80, 1088.00, 575.70, 9955.15, 3635.00, 3823.00,~
## $ MontoAdjudicado     <dbl> 400.00, 300.00, 500.00, 7951.91, 1515.00, 2025.00, 5~
## $ RendimientoPPA     <dbl> 0.09570000, 0.05225600, 0.05243900, 0.12305392, 0.07~
## $ DeclaradaDesierta   <dbl> 0, 1, 0, 1, 1, 0, 0, 1, 1, 0, 1, 0, 0, 0, 0, 0, 1~
```

```
##Transformamos nuestros datos
letras_mes_columnas <- letras_bc |>
  select(FechadeSubasta, MontoDemandado) |>
  mutate(mesFechaSubasta= month(FechadeSubasta),
         anoFechaSubasta= year(FechadeSubasta)) |>
  group_by(anoFechaSubasta, mesFechaSubasta) |>
  summarise(totalDemandado= sum(MontoDemandado)) |>
  ungroup() |>
  arrange(mesFechaSubasta)
```

'summarise()' has grouped output by 'anoFechaSubasta'. You can override using
the '.groups' argument.

```
head(letras_mes_columnas)
```

```
## # A tibble: 6 x 3
##   anoFechaSubasta mesFechaSubasta totalDemandado
##           <dbl>           <dbl>           <dbl>
## 1             2008             1             8199.
## 2             2009             1             1662.
## 3             2010             1             1772.
## 4             2011             1              456.
## 5             2012             1              282.
## 6             2013             1              709.
```