

# Date and Time Manipulation

No R não temos funções built-in para lidar com datas e hora. No entanto a biblioteca `lubridate` é muito útil para lidar com esse tipo de informação, como realizar conversões, lidar com fuso horário e etc. Afinal uma data pode ser escrita de muitas formas diferentes, e para analisar pode ser necessário alterar esse formato.

## Lubridate

As funções desse pacote são muito intuitivas, independentemente do separador entre o ano, mês e dia, basta digitar as letras **y** para ano (year), **m** para mês (month) e **d** para dia (day) na sequência em que aparece a informação e o resultado vai trazer no formato `yyyy-mm-dd`.

Vamos usar o dia 17/05/2022 para realizar os testes abaixo.

### **ymd**

```
paste(ymd("220517"),  
      ymd("20220517"),  
      ymd("2022|05|17"))
```

```
## [1] "2022-05-17 2022-05-17 2022-05-17"
```

### **mdy**

```
paste(mdy("05/17/22"),  
      mdy("05/17/2022"),  
      mdy("05|17|2022"),  
      mdy("05172022"))
```

```
## [1] "2022-05-17 2022-05-17 2022-05-17 2022-05-17"
```

Para lidar com horário, a ideia segue a mesma, utilizar as letras **h** para hora (hour), **m** para minuto (minute) e **s** para segundos (second)

## **ymd\_hms**

```
paste(ymd_hms("22-05-17-23-59-59"),  
      ymd_hms("2022-05-17 23:59:59"))
```

```
## [1] "2022-05-17 23:59:59 2022-05-17 23:59:59"
```

## **Unix Time**

É um formato numérico para representar a data, e esse número representa quantos segundos se passaram desde 1970-01-01 00:00:00 UTC. Em alguns casos essa data de origem pode ser diferente.

Aqui vamos usar uma função para resgatar a hora do sistema e converter ni formato numérico.

```
Sys.time()
```

```
## [1] "2022-05-23 21:45:25 -03"
```

Agora vamos converter em numérico para obter Unix Time

```
Sys.time() %>% ymd_hms() %>% as.numeric()
```

```
## [1] 1653342325
```

Por fim vamos pegar a data no formato unix e retornar a nossa data original.

```
Sys.time() %>% ymd_hms() %>% as.numeric() %>%  
  as.POSIXct(origin="1970-01-01", tz = "GMT")
```

```
## [1] "2022-05-23 21:45:25 GMT"
```

Aqui foi necessário utilizar a conversão para o formato POSIX (outro tipo de data), também foi necessário especificar a origem 1970-01-01, e por fim a timezone GMT que é o padrão baseado no Meridiano de Greenwich (**Greenwich Mean Time Zone**)

## Partes de uma data

Pode ser necessário extrair apenas o ano, ou o dia do ano, o dia da semana. Para cada necessidade temos uma função.

```
data <- "2022-05-17"  
paste(  
  year(data), #ano  
  yday(data), #dia do ano  
  wday(data), #dia da semana  
  wday(data, label = TRUE) #em string  
)
```

```
## [1] "2022 137 3 ter"
```

## Calcular diferença de datas

Quando usamos as funções da biblioteca lubridate, por debaixo dos panos está usando formato Unix que permite realizar cálculos.

```
ymd("2022-05-23") - ymd("2022-05-22")
```

```
## Time difference of 1 days
```

O resultado da “subtração” das datas vai indicar um período de tempo através um texto explicativo. Caso o resultado seja positivo indica que avançou no tempo, caso contrário, o resultado negativo indica que é tempo passado.

Os cálculos permitem uma granularidade a nível de horas de diferença por exemplo.

```
ymd_hms("2022-05-23 18:00:00") - ymd_hms("2022-05-23 21:00:00")
```

```
## Time difference of -3 hours
```

## duration

Há casos específicos que sabemos a duração de um evento, mas não a data final. Com essa função podemos gravar numa variável uma duração e utilizar nos cálculos.

Caso a precisão da data seja um fator importante, é preciso se limitar a no máximo duração de semanas. Unidades como mês ou anos tem maior variabilidade devido a transição de mês, ano bissexto ou até horário de verão.

```
duracao <- duration(day=1, hour=8, minute=30)
```

```
ymd("2022-05-17") + duracao
```

```
## [1] "2022-05-18 08:30:00 UTC"
```

## period

Para isso temos essa função que funciona muito bem para variações mais longas. Caso os fatores como transição de mês, ano bissexto ou até horário de verão sejam relevantes para a análise, use essa função.

```
periodo <- period(year=1, month=2, day=2)

ymd("2022-05-17") + periodo
```

```
## [1] "2023-07-19"
```

## Fuso Horário

É possível alterar o fuso horário, **nessa documentação do R** indica **esse site** para obter acesso aos diferentes fusos horário que você precisar.

```
ymd_hms("22-05-17-23-59-59", tz= "America/New_York")
```

```
## [1] "2022-05-17 23:59:59 EDT"
```

Outra possibilidade para obter essa informação de forma prática é através de uma função.

```
OlsonNames() %>% sample(10) %>% as.data.frame()
```

```
##
## 1          Asia/Dili
## 2 America/Indiana/Tell_City
## 3          Asia/Kathmandu
## 4          America/Panama
## 5          Pacific/Tahiti
## 6          Asia/Ulan_Bator
## 7 America/Rio_Branco
## 8          PRC
## 9          Asia/Dacca
## 10         Atlantic/Canary
```

Caso você não tenha certeza qual seu fuso horário, utilize a função `Sys.timezone()` para descobrir o fuso horário do seu computador.

```
Sys.timezone()
```

```
## [1] "America/Sao_Paulo"
```

Agora pensando que a data “22-05-17-23-59-59” está no fuso de São Paulo Brasil, mas queremos converter para o fuso de New York e descobrir qual o horário de lá.

```
example <- ymd_hms("22-05-17-23-59-59", tz="America/Sao_Paulo")
with_tz(example, tz= "America/New_York" )
```

```
## [1] "2022-05-17 22:59:59 EDT"
```

Uma vez que SP é GMT -3 e NY GMT -4, temos uma hora de diferença entre os fusos.