



# Corregir hora

Miriam Lerma

Marzo 2021



# Intro

En esta presentación verán como:

- Cargar datos de GPS
- Corregir la hora de acuerdo al GMT
  - Usando paquete sula
  - Usando mutate
  - Usando dmy\_hms
  - Cambiando tz
- Exportar datos

## Ustedes

- Tienes una base de datos con datos de GPS
- Tienen conocimientos básicos de R, si no te recomiendo [🔗empezar por aquí](#)
- Necesitan corregir la hora de sus GPSs



# 1. Importar datos a R

Para esta parte necesitas saber cargar tus datos, si necesitas recordar como hacerlo, te recomiendo [🔗empezar por aquí](#).

# 1.1. Cargar datos

Para cargar datos de prueba.  
Puedes instalar el paquete 'sula'

```
remotes::install_github("MiriamLL/sula")
```

## Cargar datos

```
library(sula)
GPS_raw<-GPS_raw
head(GPS_raw)
```

```
## # A tibble: 6 x 5
##   Latitude Longitude DateGMT   TimeGMT   IDs
##   <dbl>      <dbl> <chr>      <chr>      <chr>
## 1   -27.2      -109. 02/11/2017 17:05:30 GPS01
## 2   -27.2      -109. 02/11/2017 17:09:35 GPS01
## 3   -27.2      -109. 02/11/2017 17:13:50 GPS01
## 4   -27.2      -109. 02/11/2017 17:17:59 GPS01
## 5   -27.2      -109. 02/11/2017 17:22:13 GPS01
## 6   -27.2      -109. 02/11/2017 17:26:25 GPS01
```

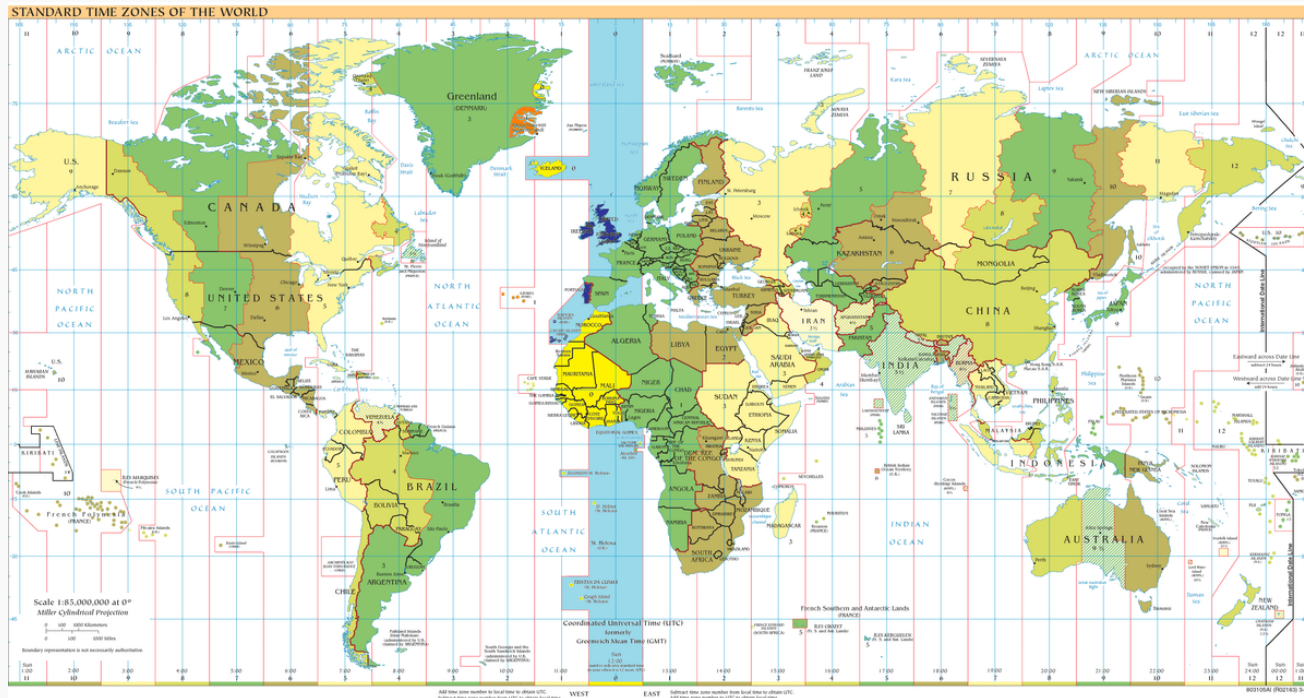


## 2. Revisar el día y la hora

Para esta parte puedes necesitar algunos básicos de como manipular data frames en R, si necesitas más información [🔗 ve aquí](#).

## 2.1. Identifica tu zona horaria

Es común que los datos de GPS se registren en GMT+0.  
Cuando usamos GPS en América Latina, tienes que identificar en que zona horaria te encuentras.



## 2.2. Día Hora

Vamos a revisar nuestros datos de día y hora.

Primero unimos las columnas de día y hora en los datos de GPS y convertimos al formato POSIXct para que R entienda que son día y hora.

Para unirlos en una columna:

```
GPS_raw$diahora<-paste(GPS_raw$DateGMT, GPS_raw$TimeGMT)
```

Para decirle que son día y hora:

```
GPS_raw$diahora<-as.POSIXct(strptime(GPS_raw$diahora,  
                                     "%d/%m/%Y %H:%M:%S"), "GMT")
```

Considera que:

- **y** es para **año**
- **m** es para **mes**
- **d** es para **día**
- **H** es para **Hora**
- **M** es para **Minuto**
- **S** es para **Segundo**

*Nota que la m para mes es en minúsculas y para Minuto en mayúsculas.*

## 2.3. Errores comunes

Verifica como tienes ordenados tus datos.

- Verifica si usas / o - para separar tus fechas, o que otro separador puede existir en tus datos.

Para el formato '21-12-2017 05:30:01'.

```
#format = "%d-%m-%Y %H:%M:%S"
```

Para el formato '2017/12/21 05:30:01'.

```
#format = "%Y/%m/%d %H:%M:%S"
```



## 2.4. Verificar datos.

Verifica que tus datos no tengan NAs ni ceros en día y hora.

```
range(GPS_raw$diahora)
```

```
## [1] "2017-11-02 17:05:30 GMT" "2017-11-27 09:48:57 GMT"
```

Si tienes **NAs**, deberas sacarlos, puedes usar la libreria tidyverse.

```
library(tidyverse)
GPS_Limpio<-GPS_raw%>%
  drop_na(diahora)
```

Si tienes ceros, puedes usar **filter** y **!=** que significa **diferente de**.

```
GPS_Limpio<-GPS_raw%>%
  filter(diahora != 0)
```



## 3. Corregir la hora

Para esta parte puedes necesitar algunos básicos de como manipular data frames en R, si necesitas más información [ve aquí](#).

## 3.1. Usando sula

Cargar la libreria.

```
library(sula)
```

Renombra el objeto.

```
Opcion1<-GPS_raw
```

Esta función corrige el tiempo de acuerdo a la zona horaria, necesitas incluir tus datos, definir la columna de hora y día, el formato en el que están y el numero de horas de diferencia.

```
GPS_gmt<-ajustar_hora(GPS_data = Opcion1,  
                      dif_hor = 5,  
                      dia_col = 'DateGMT',  
                      hora_col = 'TimeGMT',  
                      formato="%d/%m/%Y %H:%M:%S")
```

## 3.2. Usando sula

Listo!

Hora original:

```
head(GPS_raw$TimeGMT, 5)
```

```
## [1] "17:05:30" "17:09:35" "17:13:50" "17:17:59" "17:22:13"
```

Verificar hora corregida:

```
range(GPS_gmt$hora_corregida)
```

```
## [1] "2017-11-02 12:05:30 GMT" "2017-11-27 04:48:57 GMT"
```



## 3.4. Usando mutate

Cargar la libreria.

```
library(tidyverse)
```

```
GPS_raw$diahora<-paste(GPS_raw$DateGMT, GPS_raw$TimeGMT)  
GPS_raw$diahora<-as.POSIXct(strptime(GPS_raw$diahora,  
                                     "%d/%m/%Y %H:%M:%S"), "GMT")
```

Renombra el objeto.

```
Opcion2<-GPS_raw
```

Supongamos que para nuestros datos tenemos 5 horas de diferencia.

Para restar las 5 horas, creamos una columna que se llama hora\_corregida, no es necesario pero sirve para verificar que todo este correcto.

```
Opcion2<-Opcion2 %>%  
  mutate(hora_corregida = diahora - 5*60*60)
```

Nota que hora se resta a la columna diahora y las cinco horas se multiplican por los minutos y segundos.

## 3.2. Usando mutate

Listo!

```
head(Opcion2$hora_corregida, 5)
```

```
## [1] "2017-11-02 12:05:30 GMT" "2017-11-02 12:09:35 GMT"  
## [3] "2017-11-02 12:13:50 GMT" "2017-11-02 12:17:59 GMT"  
## [5] "2017-11-02 12:22:13 GMT"
```

Verificar

```
range(Opcion2$hora_corregida)
```

```
## [1] "2017-11-02 12:05:30 GMT" "2017-11-27 04:48:57 GMT"
```



## 3.3. Usando dmy\_hms

Cargar libreria.

```
library(lubridate)
```

Renombra tu objeto.

```
Opcion3<-GPS_raw
```

Convertir usando dmy\_hms

```
Opcion3$diahora<-paste(Opcion3$DateGMT,Opcion3$TimeGMT)  
Opcion3$diahora<-dmy_hms(Opcion3$diahora)
```

Restar los segundos de diferencia

```
Opcion3$hora_corregida <- Opcion3$diahora - 3600*5
```

Nota que hora se resta a la columna diahora y las cinco horas se multiplican por los minutos y segundos.

## 3.3. Usando dmy\_hms

Listo!

```
head(Opcion3$hora_corregida,5)
```

```
## [1] "2017-11-02 12:05:30 UTC" "2017-11-02 12:09:35 UTC"  
## [3] "2017-11-02 12:13:50 UTC" "2017-11-02 12:17:59 UTC"  
## [5] "2017-11-02 12:22:13 UTC"
```

Verificar

```
range(Opcion3$hora_corregida)
```

```
## [1] "2017-11-02 12:05:30 UTC" "2017-11-27 04:48:57 UTC"
```





## 3.4. Cambiando tz

Renombra tu objeto

```
GPS_raw$diahora<-paste(GPS_raw$DateGMT, GPS_raw$TimeGMT)
GPS_raw$diahora<-as.POSIXct(strptime(GPS_raw$diahora,
                                     "%d/%m/%Y %H:%M:%S"), "GMT")

Opcion4<-GPS_raw
```

Cargar paquete

```
library(lubridate)
```

El formato original es probablemente tz Europe/Amsterdam

```
Opcion4$Hora<-ymd_hms(Opcion3$diahora, tz = "Europe/Amsterdam")
```

```
head(Opcion4$Hora)
```

```
## [1] "2017-11-02 17:05:30 CET" "2017-11-02 17:09:35 CET"
## [3] "2017-11-02 17:13:50 CET" "2017-11-02 17:17:59 CET"
## [5] "2017-11-02 17:22:13 CET" "2017-11-02 17:26:25 CET"
```

## 3.4. Cambiando tz

Para cambiarlo usar `with_tz`.

```
Opcion4$hora_corregida <- with_tz(Opcion4$Hora,tzone = "America/La_Paz")
```

Listo!

```
head(Opcion4$hora_corregida,5)
```

```
## [1] "2017-11-02 12:05:30 -04" "2017-11-02 12:09:35 -04"  
## [3] "2017-11-02 12:13:50 -04" "2017-11-02 12:17:59 -04"  
## [5] "2017-11-02 12:22:13 -04"
```

Verificar

```
range(Opcion4$hora_corregida)
```

```
## [1] "2017-11-02 12:05:30 -04" "2017-11-27 04:48:57 -04"
```



## 3.4. Usando with\_tz

Para elegir la tzone tienes que conocer el **nombre de tu región** y considera el **horario de verano**.

Para ver en R la lista de nombres de las zonas horarias aceptadas usa la siguiente línea de código.

```
OlsonNames()
```

Te van a aparecer muchos resultados.

Otra opción es ver la [🔗 lista en Wikipedia](#).



## 4. Exportar

Para esta parte necesitas saber exportar tus datos, si necesitas recordar como hacerlo, te recomiendo [🔗empezar por aquí](#).

## 4.1. Exportar

Seleccionar la carpeta.

```
library(here)  
DatosFolder<-here::here("01Datos")
```

Seleccionar archivo

```
GPS_Todos<-Opcion1  
GPS_Todos<-Opcion2  
GPS_Todos<-Opcion3  
GPS_Todos<-Opcion4
```

Exportar el nuevo data frame.

```
write_csv(  
  GPS_Todos,  
  file=paste0(DatosFolder, '/GPS_Todos.csv'))
```


# 5. Contacto

Recapitulando:

- Cargar datos de GPS
- Corregir la hora de acuerdo al GMT
  - Usando paquete sula
  - Usando mutate
  - Usando dmy\_hms
  - Cambiando tz
- Exportar datos

Para dudas, comentarios y sugerencias:  
Escríbeme a [miriamjlerma@gmail.com](mailto:miriamjlerma@gmail.com)

Si quieres dar créditos puedes usar:

- Lerma M (2021) Package sula. Zenodo. <http://doi.org/10.5281/zenodo.4682898>
- Los datos de prueba vienen de esa publicación: Lerma M, Dehnhard N, Luna-Jorquera G, Voigt CC, Garthe S (2020) Breeding stage, not sex, affects foraging characteristics in masked boobies at Rapa Nui. Behavioral ecology and sociobiology 74: 149  OpenAccess

