

Lubridate Lab

Tepi

10/6/2024

Laboratorio de la libreria Lubridate

```
##cargar librerias
library(nycflights13)
library(lubridate)
library(dplyr)

glimpse(flights)

## Rows: 336,776
## Columns: 19
## $ year      <int> 2013, 2013, 2013, 2013, 2013, 2013, 2013, 2013, 2013, 2~
## $ month     <int> 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1~
## $ day       <int> 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1~
## $ dep_time  <int> 517, 533, 542, 544, 554, 554, 555, 557, 557, 558, 558, ~
## $ sched_dep_time <int> 515, 529, 540, 545, 600, 558, 600, 600, 600, 600, 600, ~
## $ dep_delay  <dbl> 2, 4, 2, -1, -6, -4, -5, -3, -3, -2, -2, -2, -2, -2, -1~
## $ arr_time  <int> 830, 850, 923, 1004, 812, 740, 913, 709, 838, 753, 849,~
## $ sched_arr_time <int> 819, 830, 850, 1022, 837, 728, 854, 723, 846, 745, 851,~
## $ arr_delay  <dbl> 11, 20, 33, -18, -25, 12, 19, -14, -8, 8, -2, -3, 7, -1~
## $ carrier   <chr> "UA", "UA", "AA", "B6", "DL", "UA", "B6", "EV", "B6", "~
## $ flight    <int> 1545, 1714, 1141, 725, 461, 1696, 507, 5708, 79, 301, 4~
## $ tailnum   <chr> "N14228", "N24211", "N619AA", "N804JB", "N668DN", "N394~
## $ origin    <chr> "EWR", "LGA", "JFK", "JFK", "LGA", "EWR", "EWR", "LGA",~
## $ dest      <chr> "IAH", "IAH", "MIA", "BQN", "ATL", "ORD", "FLL", "IAD",~
## $ air_time  <dbl> 227, 227, 160, 183, 116, 150, 158, 53, 140, 138, 149, 1~
## $ distance  <dbl> 1400, 1416, 1089, 1576, 762, 719, 1065, 229, 944, 733, ~
## $ hour      <dbl> 5, 5, 5, 5, 6, 5, 6, 6, 6, 6, 6, 6, 6, 6, 5, 6, 6, 6~
## $ minute    <dbl> 15, 29, 40, 45, 0, 58, 0, 0, 0, 0, 0, 0, 0, 0, 59, 0~
## $ time_hour <dtm> 2013-01-01 05:00:00, 2013-01-01 05:00:00, 2013-01-01 0~
```

Resuelva las siguientes preguntas:

Ejercicio 1: Convertir columnas de hora en fecha-hora

Problema: Convierte las columnas `dep_time` (hora de salida) y `arr_time` (hora de llegada) en objetos de tipo `datetime` usando `make_datetime()` de `lubridate`. Recuerda que estas columnas están en formato militar (HHMM).

Ayuda: Investiga la funcion matematica de modulo de `r`.

```
# Helper function to extract hours and minutes from military time
convert_to_time_components <- function(time) {
  hours <- time %/% 100 # Integer division to extract hours
```

```

minutes <- time %% 100 # Modulo operation to extract minutes
return(list(hours = hours, minutes = minutes))
}

# Add datetime columns to the flights dataset
flights_tidy <- flights %>%
  filter(!is.na(dep_time) & !is.na(arr_time)) %>% # Remove rows with missing times
  mutate(
    dep_datetime = make_datetime(
      year, month, day,
      hour = convert_to_time_components(dep_time)$hours,
      min = convert_to_time_components(dep_time)$minutes
    ),
    arr_datetime = make_datetime(
      year, month, day,
      hour = convert_to_time_components(arr_time)$hours,
      min = convert_to_time_components(arr_time)$minutes
    )
  )

# Display the first few rows
head(flights_tidy)

```

```

## # A tibble: 6 x 21
##   year month   day dep_time sched_dep_time dep_delay arr_time sched_arr_time
##   <int> <int> <int>   <int>         <int>      <dbl>      <int>         <int>
## 1  2013     1     1     517           515         2        830           819
## 2  2013     1     1     533           529         4        850           830
## 3  2013     1     1     542           540         2        923           850
## 4  2013     1     1     544           545        -1       1004          1022
## 5  2013     1     1     554           600        -6        812           837
## 6  2013     1     1     554           558        -4        740           728
## # i 13 more variables: arr_delay <dbl>, carrier <chr>, flight <int>,
## #   tailnum <chr>, origin <chr>, dest <chr>, air_time <dbl>, distance <dbl>,
## #   hour <dbl>, minute <dbl>, time_hour <dtm>, dep_datetime <dtm>,
## #   arr_datetime <dtm>

```

Ejercicio 2: Duracion del vuelo

Calcula el tiempo de vuelo total en minutos entre las columnas `dep_time` y `arr_time` que calculaste en el primer Ejercicio.

```

# Calcular el tiempo total de vuelo en minutos
flights_tidy <- flights_tidy %>%
  mutate(
    flight_duration = as.numeric(difftime(arr_datetime, dep_datetime, units = "mins"))
  )

# Mostrar algunas filas con la nueva columna de duración
head(flights_tidy %>% select(dep_datetime, arr_datetime, flight_duration))

```

```

## # A tibble: 6 x 3
##   dep_datetime      arr_datetime      flight_duration
##   <dtm>            <dtm>            <dbl>
## 1 2013-01-01 05:17:00 2013-01-01 08:30:00      193

```

```
## 2 2013-01-01 05:33:00 2013-01-01 08:50:00      197
## 3 2013-01-01 05:42:00 2013-01-01 09:23:00      221
## 4 2013-01-01 05:44:00 2013-01-01 10:04:00      260
## 5 2013-01-01 05:54:00 2013-01-01 08:12:00      138
## 6 2013-01-01 05:54:00 2013-01-01 07:40:00      106
```

Ejercicio 3: Extraer componentes de fechas

Extrae el día de la semana y la hora en que salieron los aviones y guardalos en las variables `dep_day_of_week` y `dep_hour`.

```
# Extraer el día de la semana y la hora de salida
flights_tidy <- flights_tidy %>%
  mutate(
    dep_day_of_week = wday(dep_datetime, label = TRUE, abbr = TRUE), # Día de la semana (abreviado)
    dep_hour = hour(dep_datetime) # Hora de salida
  )

# Mostrar algunas filas con las nuevas columnas
head(flights_tidy %>% select(dep_datetime, dep_day_of_week, dep_hour))
```

```
## # A tibble: 6 x 3
##   dep_datetime      dep_day_of_week dep_hour
##   <dtm>            <ord>          <int>
## 1 2013-01-01 05:17:00 Tue             5
## 2 2013-01-01 05:33:00 Tue             5
## 3 2013-01-01 05:42:00 Tue             5
## 4 2013-01-01 05:44:00 Tue             5
## 5 2013-01-01 05:54:00 Tue             5
## 6 2013-01-01 05:54:00 Tue             5
```

Ejercicio 4: Crear nuevas columnas con el día de la semana y la semana del año

Problema: Usando la columna `time_hour`, crea una nueva columna que indique el día de la semana y otra que indique la semana del año en la que ocurrió el vuelo.

Ayuda: Investiga la función `wday` de `lubridate`.

```
# Crear nuevas columnas con el día de la semana y la semana del año usando la columna time_hour
flights_tidy <- flights_tidy %>%
  mutate(
    flight_day_of_week = wday(time_hour, label = TRUE, abbr = FALSE), # Día de la semana completo
    flight_week_of_year = isoweek(time_hour) # Semana del año ISO
  )

# Mostrar algunas filas con las nuevas columnas
head(flights_tidy %>% select(time_hour, flight_day_of_week, flight_week_of_year))
```

```
## # A tibble: 6 x 3
##   time_hour      flight_day_of_week flight_week_of_year
##   <dtm>            <ord>          <dbl>
## 1 2013-01-01 05:00:00 Tuesday             1
## 2 2013-01-01 05:00:00 Tuesday             1
## 3 2013-01-01 05:00:00 Tuesday             1
## 4 2013-01-01 05:00:00 Tuesday             1
## 5 2013-01-01 06:00:00 Tuesday             1
```

```
## 6 2013-01-01 05:00:00 Tuesday
```

```
1
```

Ejercicio 5: Encontrar los vuelos que salieron los fines de semana

Problema: Filtra los vuelos que despegaron un sábado o domingo y devuelve el total de vuelos en fines de semana.

```
# Filtrar los vuelos que salieron durante el fin de semana (sábado o domingo)
weekend_flights <- flights_tidy %>%
  filter(flight_day_of_week %in% c("Saturday", "Sunday"))

# Contar el total de vuelos en fines de semana
total_weekend_flights <- nrow(weekend_flights)

weekend_flights_selected <- weekend_flights %>%
  select(time_hour, flight_day_of_week)

# Mostrar el resultado
total_weekend_flights
```

```
## [1] 83479
```

```
weekend_flights_selected
```

```
## # A tibble: 83,479 x 2
##   time_hour      flight_day_of_week
##   <dtm>         <ord>
## 1 2013-01-05 23:00:00 Saturday
## 2 2013-01-05 22:00:00 Saturday
## 3 2013-01-05 05:00:00 Saturday
## 4 2013-01-05 05:00:00 Saturday
## 5 2013-01-05 05:00:00 Saturday
## 6 2013-01-05 05:00:00 Saturday
## 7 2013-01-05 06:00:00 Saturday
## 8 2013-01-05 06:00:00 Saturday
## 9 2013-01-05 06:00:00 Saturday
## 10 2013-01-05 06:00:00 Saturday
## # i 83,469 more rows
```