

Tarea10

Yimmy Eman

2022-07-08

```
library(tidyverse)

## -- Attaching packages ----- tidyverse 1.3.1 --

## v ggplot2 3.3.6      v purrr 0.3.4
## v tibble 3.1.7       v dplyr 1.0.9
## v tidyr 1.2.0        v stringr 1.4.0
## v readr 2.1.2        v forcats 0.5.1

## -- Conflicts ----- tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()     masks stats::lag()

library(nycflights13)
```

Pregunta 1

El dataset de vuelos tiene dos variables, `dep_time` y `sched_dep_time` muy útiles pero difíciles de usar por cómo vienen dadas al no ser variables continuas. Fíjate que cuando pone 559, se refiere a que el vuelo salió a las 5:59...

Convierte este dato en otro más útil que represente el número de minutos que han transcurrido desde media noche.

```
transmute(flights,
  dep_time,
  new_dep_time = 60 * dep_time %/% 100 + dep_time %% 100,
  sched_dep_time,
  new_shed_dep_time = 60 * sched_dep_time %/% 100 + sched_dep_time %% 100
)
```

```
## # A tibble: 336,776 x 4
##   dep_time new_dep_time sched_dep_time new_shed_dep_time
##   <int>      <dbl>         <int>         <dbl>
## 1     517         317           515           60
## 2     533         333           529           60
## 3     542         342           540           60
## 4     544         344           545           60
## 5     554         354           600           60
```

```
## 6      554      354      558      60
## 7      555      355      600      60
## 8      557      357      600      60
## 9      557      357      600      60
## 10     558      358      600      60
## # ... with 336,766 more rows
```

Pregunta 2

Compara las variables `air_time` contra `arr_time - dep_time`.

- ¿Qué esperas ver?
- ¿Qué ves realmente?
- ¿Se te ocurre algo para mejorarlo y corregirlo?

```
transmute(flights,air_time,
          new_dep_time = 60*dep_time%%100+dep_time%%100,
          new_arr_time = 60*arr_time%%100+arr_time%%100,
          new_air_time = new_arr_time-new_dep_time)
```

```
## # A tibble: 336,776 x 4
##   air_time new_dep_time new_arr_time new_air_time
##   <dbl>      <dbl>      <dbl>      <dbl>
## 1      227         317         510         193
## 2      227         333         530         197
## 3      160         342         563         221
## 4      183         344         604         260
## 5      116         354         492         138
## 6      150         354         460         106
## 7      158         355         553         198
## 8       53         357         429          72
## 9      140         357         518         161
## 10     138         358         473         115
## # ... with 336,766 more rows
```

Pregunta 3

Compara los valores de `dep_time`, `sched_dep_time` y `dep_delay`. Cómo deberían relacionarse estos tres números? Compruébalo y haz las correcciones numéricas que necesitas.

```
transmute(flights,
          new_dep_time = 60*dep_time%%100+dep_time%%100,
          new_sched_dep_time = 60*sched_dep_time%%100+sched_dep_time%%100,
          new_delay = new_dep_time-new_sched_dep_time,
          dep_delay,
          new_delay==dep_delay)
```

```
## # A tibble: 336,776 x 5
##   new_dep_time new_sched_dep_time new_delay dep_delay 'new_delay == dep_delay'
```

```
##           <dbl>           <dbl>      <dbl>      <dbl> <dbl> <lgl>
## 1           317           315          2          2 TRUE
## 2           333           329          4          4 TRUE
## 3           342           340          2          2 TRUE
## 4           344           345         -1         -1 TRUE
## 5           354           360         -6         -6 TRUE
## 6           354           358         -4         -4 TRUE
## 7           355           360         -5         -5 TRUE
## 8           357           360         -3         -3 TRUE
## 9           357           360         -3         -3 TRUE
## 10          358           360         -2         -2 TRUE
## # ... with 336,766 more rows
```

Pregunta 4

Usa una de las funciones de ranking para quedarte con los 10 vuelos más retrasados de todos.

```
arrange(mutate(flights, r_delay = min_rank(dep_delay)), r_delay)[1:10,]
```

```
## # A tibble: 10 x 20
##   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    12     7    2040         2123         -43     40         2352
## 2  2013     2     3    2022         2055         -33    2240         2338
## 3  2013    11    10    1408         1440         -32    1549         1559
## 4  2013     1    11    1900         1930         -30    2233         2243
## 5  2013     1    29    1703         1730         -27    1947         1957
## 6  2013     8     9     729         755         -26    1002          955
## 7  2013    10    23    1907         1932         -25    2143         2143
## 8  2013     3    30    2030         2055         -25    2213         2250
## 9  2013     3     2    1431         1455         -24    1601         1631
## 10 2013     5     5     934         958         -24    1225         1309
## # ... with 12 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>, r_delay <int>
```

Pregunta 5

Aunque la ejecución te dé una advertencia, qué resultado te da la operación 1:6 + 1:20

```
1:6 + 1:20
```

```
## Warning in 1:6 + 1:20: longer object length is not a multiple of shorter object
## length
```

```
## [1] 2 4 6 8 10 12 8 10 12 14 16 18 14 16 18 20 22 24 20 22
```

Pregunta 6

Además de todas las funciones que hemos dicho, las trigonométricas también son funciones vectoriales que podemos usar para hacer transformaciones con `mutate`. Investiga cuales trae R y cual es la sintaxis de cada una de ellas.

seno, coseno y la tangente y sus vertientes hiperbólicas.