

Tarea21

Yimmy Eman

2022-07-16

Pregunta 1

¿Qué ocurre si procesas un string con formato de fecha inválido? `ymd(c("2018-06-08", "Antonio"))`

```
ymd(c("2018-06-08", "Antonio"))
```

```
## Warning: 1 failed to parse.
```

```
## [1] "2018-06-08" NA
```

Pregunta 2

¿Cuál es la función del argumento `tzzone` en la función `today()`. ¿Es importante configurarlo?

```
today()
```

```
## [1] "2022-07-16"
```

```
today("UTC")
```

```
## [1] "2022-07-16"
```

```
today(tzzone = "America/Caracas")
```

```
## [1] "2022-07-16"
```

Pregunta 3

Usa la función adecuada de `lubridate` para procesar las siguientes:

```
d1 <- "January 1, 2018"
d2 <- "2015-May-19"
d3 <- "08-Jun-2018"
d4 <- c("August 19 (2015)", "July 1 (2015)")
d5 <- "12/30/18" # 30 de Diciembre de 2018
```

```
mdy(d1)
```

```
## [1] "2018-01-01"
```

```
ymd(d2)
```

```
## [1] "2015-05-19"
```

```
dmy(d3)
```

```
## [1] "2018-06-08"
```

```
mdy(d4)
```

```
## [1] "2015-08-19" "2015-07-01"
```

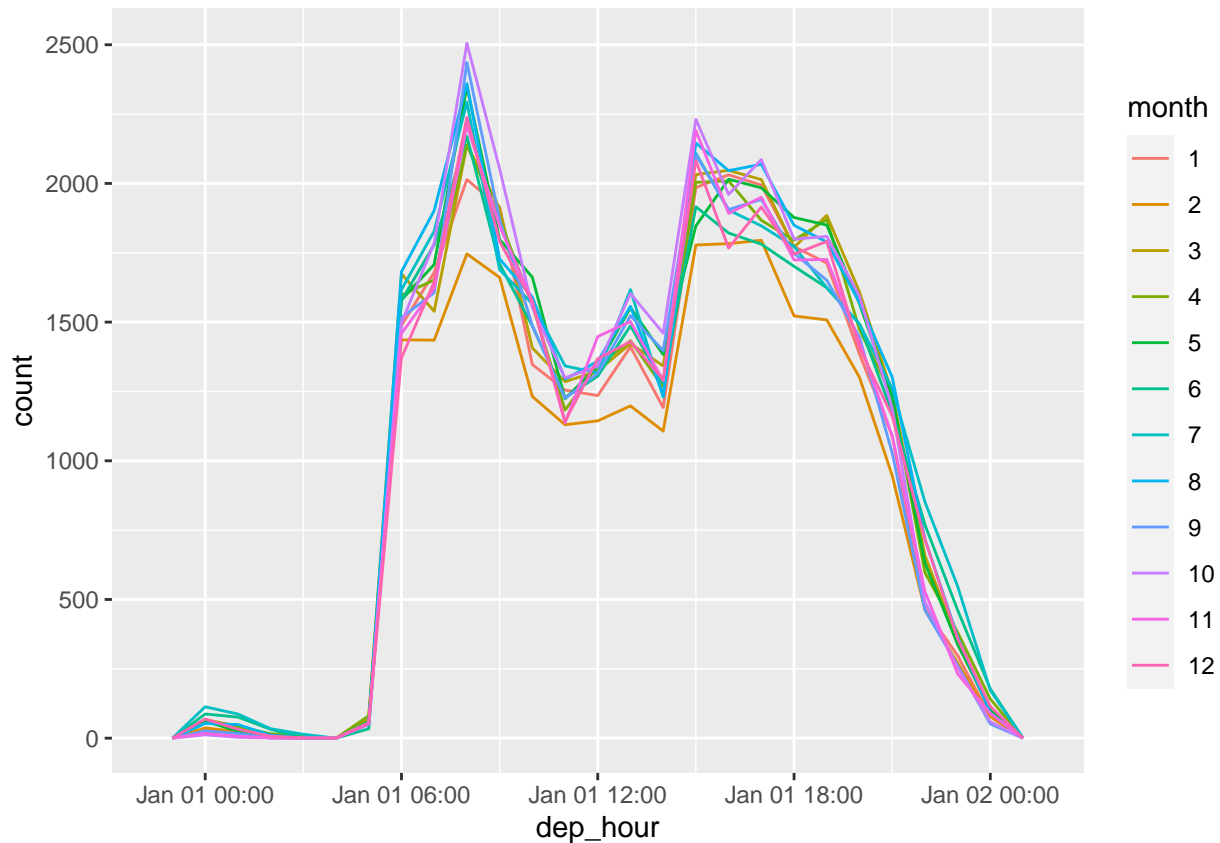
```
mdy(d5)
```

```
## [1] "2018-12-30"
```

Pregunta 4

1. Indica cómo cambia la distribución de horas de vuelo de un día a lo largo de todo el año.

```
make_datetime_100 <- function(year, month, day, time) {  
  make_datetime(year, month, day, time %% 100, time %% 100)  
}  
  
flights_dt <- flights %>%  
  filter(!is.na(dep_time), !is.na(arr_time)) %>%  
  mutate(  
    dep_time = make_datetime_100(year, month, day, dep_time),  
    arr_time = make_datetime_100(year, month, day, arr_time),  
    sched_dep_time = make_datetime_100(year, month, day, sched_dep_time),  
    sched_arr_time = make_datetime_100(year, month, day, sched_arr_time)  
  ) %>%  
  select(origin, dest, ends_with("delay"), ends_with("time"))  
  
flights_dt %>%  
  filter(!is.na(dep_time)) %>%  
  mutate(dep_hour = update(dep_time, yday = 1)) %>%  
  mutate(month = factor(month(dep_time))) %>%  
  ggplot(aes(dep_hour, color = month)) +  
  geom_freqpoly(binwidth = 60 * 60)
```



2. Compara `dep_time`, `sched_dep_time` y `dep_delay`. ¿Son constantes? Explica qué conclusiones puedes sacar.

```
flights_dt %>%
  mutate(dep_time_ = sched_dep_time + dep_delay * 60) %>%
  filter(dep_time_ != dep_time) %>%
  select(dep_time_, dep_time, sched_dep_time, dep_delay)
```

```
## # A tibble: 1,205 x 4
##   dep_time_      dep_time      sched_dep_time  dep_delay
##   <dtm>         <dtm>         <dtm>         <dbl>
## 1 2013-01-02 08:48:00 2013-01-01 08:48:00 2013-01-01 18:35:00    853
## 2 2013-01-03 00:42:00 2013-01-02 00:42:00 2013-01-02 23:59:00     43
## 3 2013-01-03 01:26:00 2013-01-02 01:26:00 2013-01-02 22:50:00    156
## 4 2013-01-04 00:32:00 2013-01-03 00:32:00 2013-01-03 23:59:00     33
## 5 2013-01-04 00:50:00 2013-01-03 00:50:00 2013-01-03 21:45:00    185
## 6 2013-01-04 02:35:00 2013-01-03 02:35:00 2013-01-03 23:59:00    156
## 7 2013-01-05 00:25:00 2013-01-04 00:25:00 2013-01-04 23:59:00     26
## 8 2013-01-05 01:06:00 2013-01-04 01:06:00 2013-01-04 22:45:00    141
## 9 2013-01-06 00:14:00 2013-01-05 00:14:00 2013-01-05 23:59:00     15
## 10 2013-01-06 00:37:00 2013-01-05 00:37:00 2013-01-05 22:30:00    127
## # ... with 1,195 more rows
```

3. Compara `air_time` con la duración entre el tiempo de salida y de llegada. Explica tus conclusiones teniendo en cuenta la localización del aeropuerto.

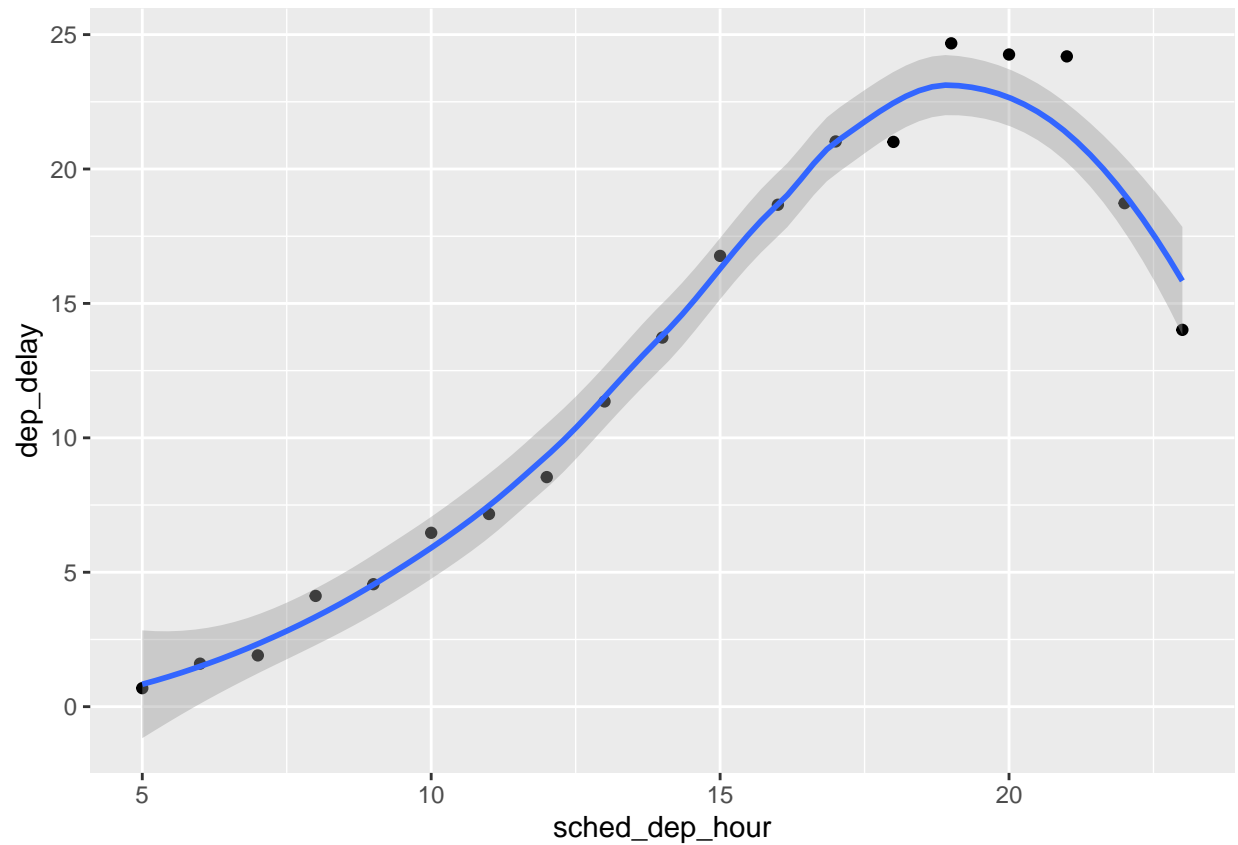
```
flights_dt %>%
  mutate(
    flight_duration = as.numeric(arr_time - dep_time),
    air_time_mins = air_time,
    diff = flight_duration - air_time_mins
  ) %>%
  select(origin, dest, flight_duration, air_time_mins, diff)
```

```
## # A tibble: 328,063 x 5
##   origin dest flight_duration air_time_mins diff
##   <chr> <chr>          <dbl>          <dbl> <dbl>
## 1 EWR   IAH             193             227   -34
## 2 LGA   IAH             197             227   -30
## 3 JFK   MIA             221             160    61
## 4 JFK   BQN             260             183    77
## 5 LGA   ATL             138             116    22
## 6 EWR   ORD             106             150   -44
## 7 EWR   FLL             198             158    40
## 8 LGA   IAD              72              53    19
## 9 JFK   MCO             161             140    21
## 10 LGA  ORD             115             138   -23
## # ... with 328,053 more rows
```

4. ¿Cómo cambia el promedio del retraso a lo largo del día? ¿Tienes que usar `dep_time` o `sched_dep_time`? ¿Por qué?

```
flights_dt %>%
  mutate(sched_dep_hour = hour(sched_dep_time)) %>%
  group_by(sched_dep_hour) %>%
  summarise(dep_delay = mean(dep_delay)) %>%
  ggplot(aes(y = dep_delay, x = sched_dep_hour)) +
  geom_point() +
  geom_smooth()
```

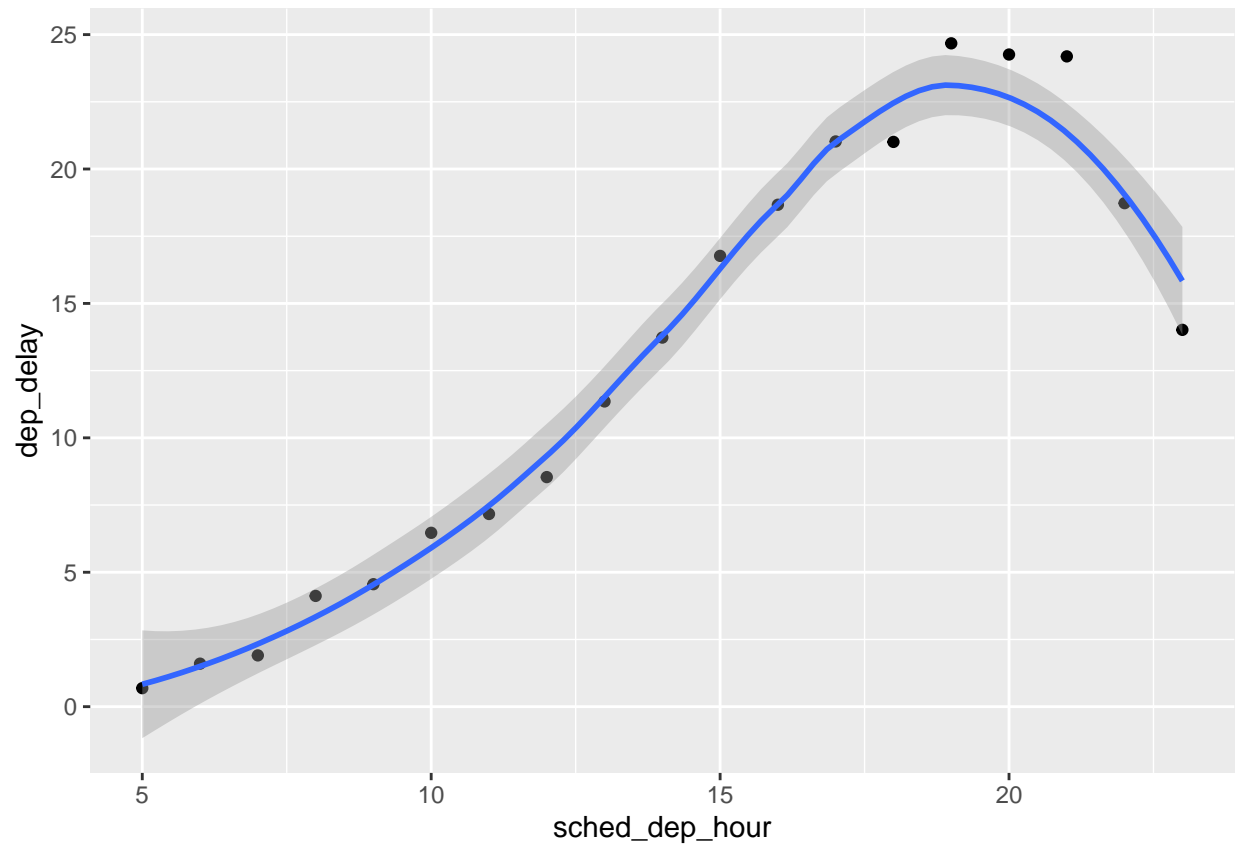
```
## 'geom_smooth()' using method = 'loess' and formula 'y ~ x'
```



5. ¿Qué día de la semana tenemos que volar para minimizar el riesgo de retraso?

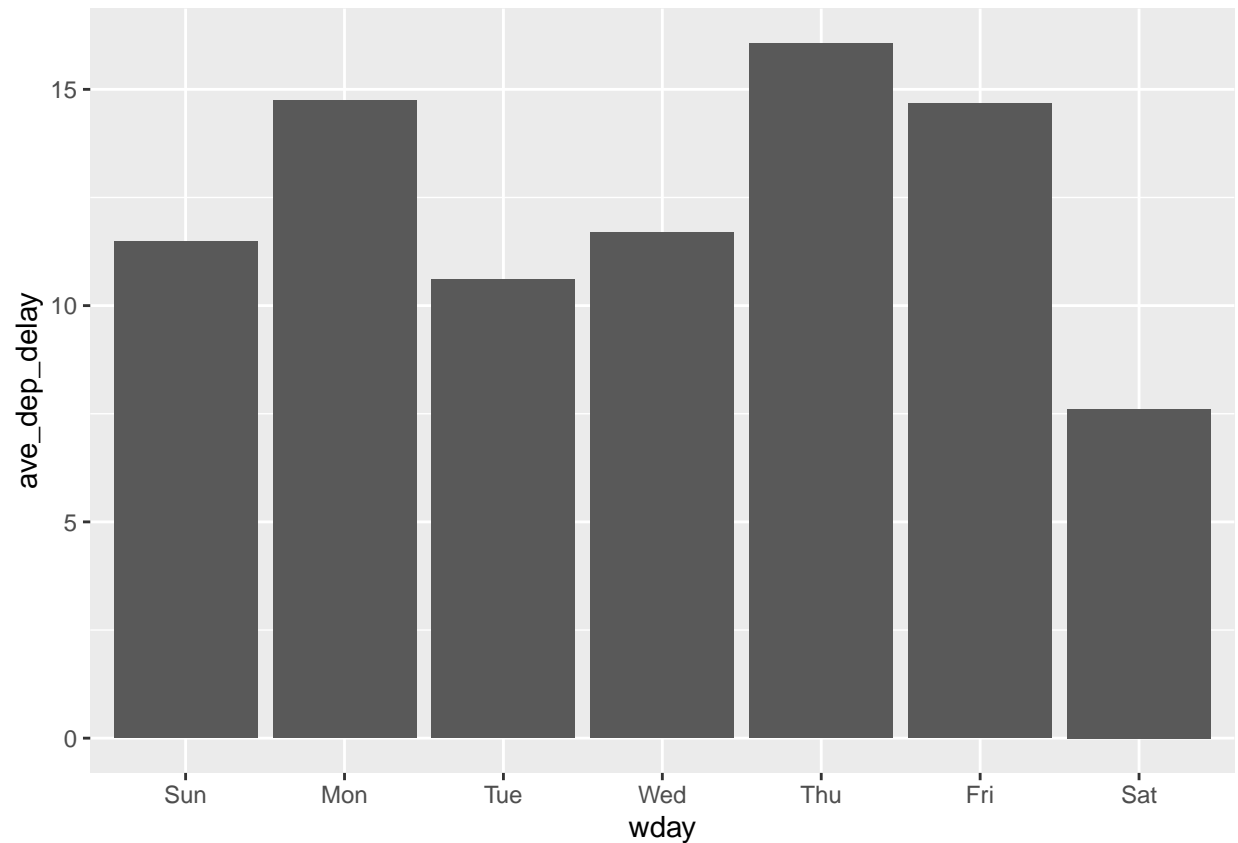
```
flights_dt %>%
  mutate(sched_dep_hour = hour(sched_dep_time)) %>%
  group_by(sched_dep_hour) %>%
  summarise(dep_delay = mean(dep_delay)) %>%
  ggplot(aes(y = dep_delay, x = sched_dep_hour)) +
  geom_point() +
  geom_smooth()
```

```
## 'geom_smooth()' using method = 'loess' and formula 'y ~ x'
```



6. ¿En qué se parecen las distribuciones de `diamonds`, `carats`, `flight`, `ssched_dep_time`

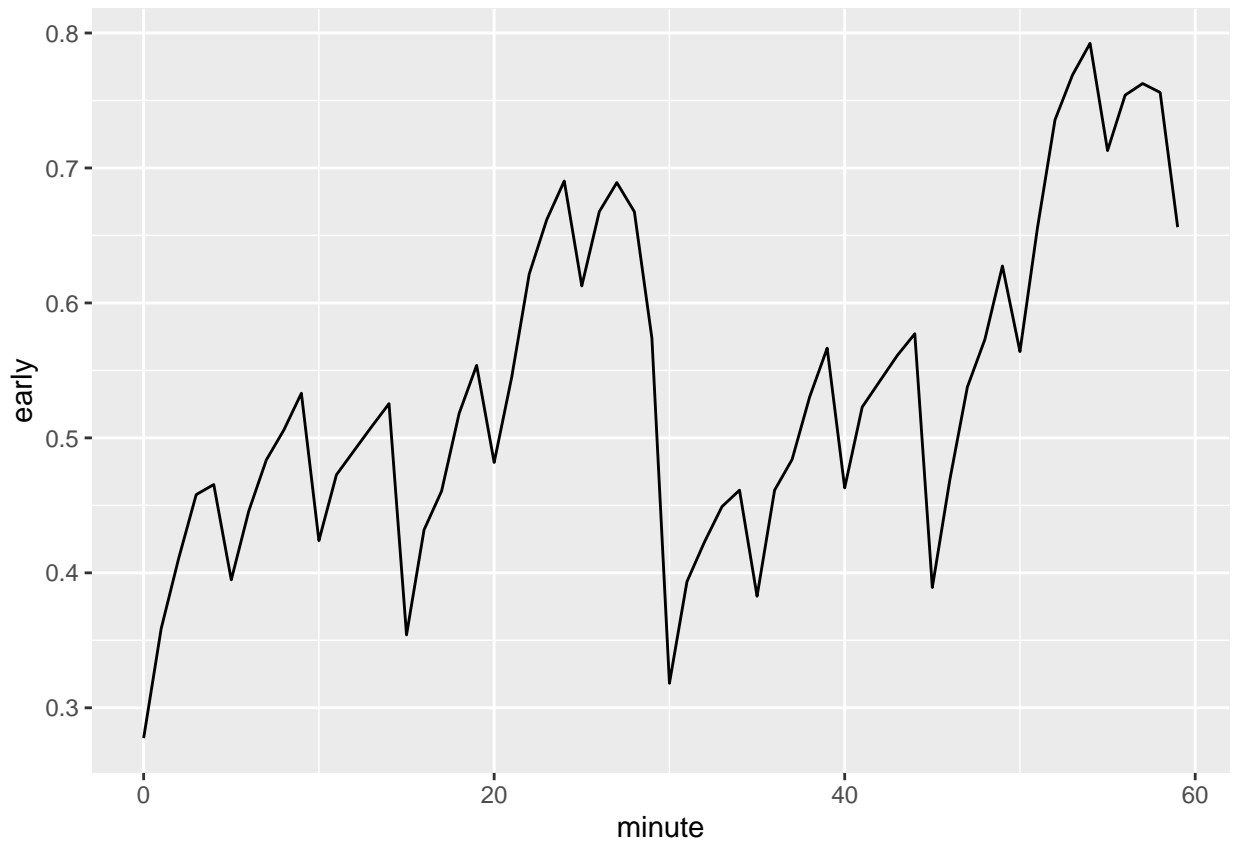
```
flights_dt %>%
  mutate(wday = wday(dep_time, label = TRUE)) %>%
  group_by(wday) %>%
  summarize(ave_dep_delay = mean(dep_delay, na.rm = TRUE)) %>%
  ggplot(aes(x = wday, y = ave_dep_delay)) +
  geom_bar(stat = "identity")
```



Pregunta 5

Confirma la hipótesis de que los vuelos que salen más temprano en los minutos 20-30 y 50-60 son debidos a que hay vuelos programados que han salido antes de y media y de en punto. Ayúdate de una variable binaria para decidir si un vuelo ha sido o no retrasado.

```
flights_dt %>%
  mutate(minute = minute(dep_time),
         early = dep_delay < 0) %>%
  group_by(minute) %>%
  summarise(
    early = mean(early, na.rm = TRUE),
    n = n()) %>%
  ggplot(aes(minute, early)) +
  geom_line()
```



Pregunta 6

1. Investiga por qué existe la función `months()` pero no `dmonths()`
2. Explica para humanos corrientes la función `days(overnight * 1)` a alguien que acaba de iniciarse en R.

La variable durante la noche es igual a VERDADERO o FALSO. Si es un vuelo nocturno, este se convierte en 1 día, y si no, entonces nocturno = 0, y no se agregan días a la fecha.

3. Crea un vector de fechas que empiecen el primer día de cada mes de 2012.

```
ymd("2012-01-01") + months(0:11)
```

```
## [1] "2012-01-01" "2012-02-01" "2012-03-01" "2012-04-01" "2012-05-01"
## [6] "2012-06-01" "2012-07-01" "2012-08-01" "2012-09-01" "2012-10-01"
## [11] "2012-11-01" "2012-12-01"
```

4. Crea un vector de fechas que empiecen el primer día de cada mes del año actual.

```
floor_date(today(), unit = "year") + months(0:11)
```

```
## [1] "2022-01-01" "2022-02-01" "2022-03-01" "2022-04-01" "2022-05-01"
## [6] "2022-06-01" "2022-07-01" "2022-08-01" "2022-09-01" "2022-10-01"
## [11] "2022-11-01" "2022-12-01"
```


5. Escribe una función que dado un cumpleaños (una fecha) devuelva cuantos años tienes.

```
age <- function(bday) {  
  (bday %--% today()) %/% years(1)  
}  
age(ymd("1988-11-27"))
```

```
## [1] 33
```

6. ¿Por qué no funciona (today() %-% (today() + years(1)) / months(1) ?

```
as.Date("2016-02-29") + years(1)
```

```
## [1] NA
```